DATA-DRIVEN AIRCRAFT ASSIGNMENT AND STOCHASTIC MODELS FOR SERVICE SYSTEMS

Wei Liu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2022

Approved by:

Nilay Argon

Vinayak Deshpande

Vidyadhar Kulkarni

Yao Li

Quoc Tran-Dinh

ABSTRACT

WEI LIU: DATA-DRIVEN AIRCRAFT ASSIGNMENT AND STOCHASTIC MODELS FOR SERVICE SYSTEMS
(Under the direction of Vinayak Deshpande and Vidyadhar Kulkarni)

This dissertation consists of two parts: data-driven aircraft assignment and stochastic models for service systems. In the first part, we propose a data-driven approach to reduce the delay propagation by optimizing the assignment between incoming and outgoing flights flown by an airline. There are two projects in this part. In the first project, we consider the aircraft assignment problem at a single airport. We propose a data-driven approach to estimate the assignment cost by considering covariates including scheduled arrival time, originating airport and aircraft type of the flights. We conclude that the stochastic assignment derived from this data-driven approach significantly outperforms the actual assignment. In the second project in this part, we extend the previous project to a network of airports by optimizing the assignment between incoming and outgoing flights at each airport in the network. We propose a similar data-driven approach to estimate the assignment costs at each airport, and show that our approach performs better than the benchmark policies.

In the second part, we consider the stochastic models for service systems. There are two projects in this part as well. In the first project, we consider a joint staffing and admission control problem under minimal, partial and full information cases. We compare the profit under different information cases over the parameter space in detail. In the second project, we consider the joint admission and service rate control problem for a general reward structure under an unobservable (minimal information case) single server queueing system. We show that when the per unit service cost is less than or equal to a critical value, it is optimal to admit all the customers, otherwise, it is optimal to admit none. We show that this socially optimal policy induces the customers to behave in a socially optimal way with self-regulation.

ACKNOWLEDGEMENTS

First of all, I would thank my advisors, Dr. Vinayak Deshpande and Dr. Vidyadhar Kulkarni, for their guidance and encouragement during my PhD study. Without their unreserved help and support, this dissertation would not have been possible. I would also express my gratitude to my dissertation committee members, Dr. Nilay Argon, Dr. Yao Li and Dr. Quoc Tran-Dinh. They gave me valuable comments and suggestions on my research. I appreciate their great help.

I would also express my gratitude to the faculty members and staff in our department for their help and support in these five years. With their great help, I have learned and improved a lot during my PhD study. After these five years' study in the department, I have grown as a researcher and an instructor.

I would also express many thanks to my friends. They offered me valuable suggestions when I face some difficult questions. We also attended lots of interesting activities together, for example, badminton. They make my life more colorful.

Finally, I would express my deepest gratitude to my parents, Yong Liu and Xiuxia Dong. Without their love and support, I would not have the opportunity to pursue a PhD in US.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

This thesis consists two parts. In the first part, we consider the data-driven aircraft assignment to minimize the delay propagation in the airline industry. We consider the assignment first at a single airport and then in a network of airports. In the second part, we consider the stochastic models for service systems. Specifically, we first consider a joint staffing and admission control problem for a binary reward structure under different levels of information, and then analyze the joint admission and service rate control under a general reward structure for an unobservable queue. The terms used in the rest of this chapter will be defined precisely in the respective chapters.

In the first part, we propose a data-driven approach to reduce the delay propagation by optimizing the assignment between incoming and outgoing flights flown by an airline. There are two projects in this chapter. In the first project, we consider the aircraft assignment problem at a single airport. We consider both deterministic and stochastic arrival times. In the deterministic case, we consider two different objective functions: nonnegative and signed delays. We show the optimality of the First-In, First-Out (FIFO) assignment for these objectives. In the stochastic case, we propose the revised FIFO (rFIFO) and stochastic assignments based on the mean and distribution of arrival delay, respectively. The central component to derive the rFIFO and stochastic assignments is the estimation of arrival delay distribution. We provide a data-driven approach to estimate the arrival delay distribution by considering covariates like scheduled arrival time, originating airport and aircraft type of the flights, which is further used to estimate assignment cost. We derive the rFIFO and stochastic assignments based on the estimation of the assignment cost. Then we compare the FIFO assignment obtained from the deterministic setting with the rFIFO and stochastic assignments derived from the data-driven approach using the real-world data. We show that the rFIFO and stochastic assignments offer a verifiable improvement compared to the FIFO assignment, and the stochastic assignment works the best by using the real data of Delta

Airlines at Atlanta airport. We conclude that the stochastic assignment would have saved Delta 6.5 million dollars in delay costs annually.

In the second project in this part, we optimize the assignment between incoming and outgoing flights at each airport in the network. We propose an iterative algorithm under the deterministic, mixed and stochastic cases to solve the aircraft assignment problem. The computational results indicate that the iterative algorithm brings a lower total expected propagated delay compared to the previous algorithms, and reduce the computation time substantially. We also show that the assignment derived from the deterministic and mixed cases underestimate the total expected propagated delay significantly. An important component of the aircraft assignment problem over the entire network of flights is the estimation of primary delay distribution. We propose a data-driven approach under the deterministic, mixed and stochastic cases to estimate the primary delay distribution, which is further used to derive the aircraft assignment for future operations. We show that the data-driven approach under the stochastic case performs better than the data-driven approach under the deterministic and mixed cases. Finally, we compare the assignment derived from the data-driven approach under the stochastic case with two benchmark policies on two networks operated by one major airline. The result shows that our approach outperforms the benchmark policies in terms of the total expected propagated delay without degrading the percentage of delayed flights.

In the second part, we consider the stochastic models for service systems. There are two projects in this part as well. In the first project, we consider a joint staffing and admission control problem with three different levels of information. For the joint staffing and admission control problem, we consider it into two steps, namely admission control and staffing problem. For the admission control problem, we aim to propose an admission policy to maximize the reward rate, which is earned by the service provider if the incoming customer's queueing time is within a predefined time. Since there is a cost for each server per unit time, we get the optimal staffing level by maximizing the profit rate, defined as the revenue rate minus the cost incurred by each server in staffing design problem. The joint staffing and admission control problem would be affected by the information we have. We consider three different levels of information, namely, minimal, partial and full information cases. In the minimal information case, we only know the system parameters, such as arrival rate, service rate and the predefined time, and we do not know anything else about

the state of the system. In the partial information case, we know the number of customers in the system at the time of arrival in addition to the information known in the minimal information case. In the full information case, we know the exact queueing time of incoming customers in addition to the information known in the minimal information case. We show that the trends of optimal admission control policy under the minimal and partial information cases are similar, which are very different from that under the full information case. We also show the difference in profit under different information cases within the parameter space. Our model can not only help the system manager decide the optimal admission control and staffing policy based on the information he has about the system, but also help the system manager realize the potential improvement in profit if he can get additional information.

In the second project in this part, we consider the joint admission and service rate control problem for a general reward structure for an unobservable queue. The system operator has two controls: the admission probability, and the average service rate. We prove the optimal admission and service rate control policy when the general reward structure satisfying certain conditions. Then we give four different kinds of reward structures satisfying such conditions. We further show that our proposed reward structure could make sure the customer can behave in a socially optimal way. We also show that the centralized decision, Stackelberg equilibrium and Nash equilibrium are equivalent in deriving the optimal admission and service rate control policy. It implies that our proposed optimal policy is applicable to a very general setting.

CHAPTER 2

Introduction to Data-Driven Aircraft Assignment

Flight on-time performance, OTP in short, is a widely accepted metric that airlines use globally to demonstrate punctuality of their flight networks. Airlines are compared on their OTPs and the metric often serves as a potential service differentiator for marketing the brand to air travelers (Official Airline Guide 2020). The U.S. Department of Transportation (DOT) considers a flight to be delayed when its actual arrival time is at least 15 minutes later than its scheduled arrival time. In 2018, 81.3% of flights arrived on time in the United States based on the DOT definition of on-time arrivals. Counting only those flights whose actual arrival time is no later than the scheduled arrival time further reduces the on-time performance for arrivals to 65.1% in 2018.

(Ball et al., 2010) estimate that the passenger time lost due to schedule buffers, delayed flights, flight cancellations, and missed connections costed the US economy approximately $16.7 billion in 2007. The $8.3 billion direct costs to airlines consisted of increased expenses for crew, fuel, and maintenance, among others. A report by the Joint Economic Committee of the U.S. Congress (Schumer and Maloney 2008) estimated the total cost of flight delays to the U.S. economy was as much as $41 billion in 2007. Clearly, flight delays have a significant impact on the U.S. economy.

Airlines typically blame flight delays on a number of external factors that are out of their control. Flight delays have been attributed to several causes such as weather conditions, airport congestion, airspace congestion, use of smaller aircraft by airlines, etc. The Bureau of Transportation Statistics (BTS) provides a breakdown of flight delays in the U.S. into five categories: (i) Air Carrier delay, (ii) Aircraft Arriving Late, (iii) National Aviation System delay, (iv) Security delay, and (v) Extreme Weather delay. It is important to note that the largest contributor of flight delays in recent years has been the propagated delay due to late arriving aircraft. Close to 40% of all delays have been attributed to aircraft arriving late by BTS during 2011-2019. Flight delays are caused due to two factors: (i) the randomness in the intrinsic travel time for a scheduled flight (which is the travel time excluding propagated delays), and (ii) the propagation of this randomness through the air-travel

network and infrastructure. While a large portion of the intrinsic randomness in travel time of a flight can be attributed to factors outside an airline's control such as weather, the propagation of this randomness in an airline's network is largely driven by factors within an airlines' control such as aircraft routing and flight scheduling decisions.

The airline schedule development process consists of four phases (Deshpande and Arıkan 2012): (1) service planning, (2) schedule generation, (3) resource allocation, and (4) execution scheduling. The service planning phase is conducted by the marketing group with the goal of creating a set of services that an airline will offer in each market. This usually consists of the frequency of flights offered in each market and also usually includes desired time windows (e.g., 5 p.m.–6 p.m.) and aircraft types (e.g., wide body, narrow body, long range, etc.). The scheduling group takes this service plan and develops the actual passenger schedules by considering aggregate constraints such as the total number of available aircraft and flight crew. Note that each passenger schedule includes the exact departure and arrival time of each flight, and hence the scheduled block-time decision for each flight is made at this stage (i.e., schedule generation phase). The passenger schedules then become an input to various specific resource allocation decisions that are usually the responsibility of the operations group (i.e., resource allocation phase). For example, aircraft with specific tail numbers are assigned to appropriate aircraft rotations by taking into consideration various constraints such as maintenance requirements. Finally, the execution scheduling phase involves implementing the developed schedule by taking schedule deviations (irregular operations) into account.

Thus, propagation of delays in an airlines' network can be reduced through proper scheduling decisions, as well as resource allocation decisions in the planning phase. Traditional schedule planning and routing models solve a deterministic optimization problem that assumes that flight times are known and fixed (Barnhart and Cohn 2004). Recent approaches that incorporate uncertainty in the planning process captures the stochastic nature of travel times. One approach is to change flight schedules to make them more robust by adding buffers in the schedule that can absorb delays in arriving flights. A second approach is to create aircraft assignments for a fixed flight schedule to minimize delay propagation. The aircraft routing problem is to assign tail numbers on scheduled arriving flights at an airport to scheduled departing flights from the same airport with the objective of minimizing propagated delays. We focus on this second approach in our research.

There are two perspectives within this specific category. Previous research, such as (Dunbar et al., 2012) and (Dunbar et al., 2014), considers the aircraft routing problem from a string-based perspective, i.e., selecting the strings (a sequence of connected flight legs assigned to an aircraft that begins and ends at maintenance stations) that minimizes the total delay propagation. However, we consider the problem from a leg-based perspective, i.e., optimizing the assignment of aircraft tail numbers between incoming and outgoing flights at each airport in the network to minimize the total delay propagation. We call this specific assignment problem as a *aircraft assignment problem*. If the arrival delays are known precisely (unlikely in practice), then the aircraft assignment problem can be formulated as a traditional assignment model, which can be solved very easily. However, since arrival delays are unknown during the planning phase, the aircraft assignment problem can be formulated as a stochastic assignment model with random assignment costs. The challenge in solving this problem arises in estimating the (stochastic) assignment costs associated with assigning the tail number of an arriving flight to a departing flight at that airport since the assignment cost depends on arrival delays.

We propose a data-driven approach to estimate the arrival delay distribution for the aircraft assignment problem at a single airport in the first project, and propose a data-driven approach to estimate the primary delay distribution for the aircraft assignment problem at each airport in the second project. In the first project, we estimate the assignment costs by using empirical observations of arrival delays from prior years' flight records to compute the estimated propagated delay associated with connecting an arriving aircraft tail number to a departing flight. We propose a data-driven clustering method to account for factors such as originating airport of an arriving flight, scheduled time of arrival, and aircraft type to translate the empirical observations of prior years' arrival delays to the assignment costs for our model. These assignment costs are used to test the performance of several aircraft assignment policies using a hold-out sample data set.

In the second project, we propose a data-driven approach to estimate the assignment costs at each airport for the aircraft assignment problem. We first propose an iterative algorithm under the deterministic, mixed and stochastic cases to solve the aircraft assignment problem. An important component of the aircraft assignment problem over the entire network of flights is the estimation of primary delay distribution. We propose a data-driven approach under the deterministic, mixed and stochastic cases to estimate the primary delay distribution, which is further used to derive

the aircraft assignment for future operations. Finally, we compare the assignment derived from the data-driven approach under the stochastic case with two benchmark policies on two networks operated by one major airline. The result shows that our approach outperforms the benchmark policies in terms of the total expected propagated delay without degrading the percentage of delayed flights.

CHAPTER 3

Literature Review to Data-Driven Aircraft Assignment

Academic literature has examined two approaches to minimize airline propagated delays and disruptions: (i) *ex-post* plans to minimize the impact of observed actual disruptions, (ii) *ex-ante* plans to reduce potential delays and disruptions. *Ex-post* airline delay research includes (Yen and Birge, 2006) , (Froyland et al., 2013) and (Wei and Vaze, 2018). (Lee et al., 2020) is the first to propose a joint *ex-post* and *ex-ante* approach to optimize the recovery decisions in response to realized disruptions and in anticipation of future disruptions. Our focus is on the *ex-ante* plans which are typically made in the schedule generation and resource allocation phase of the airline scheduling process. There are two *ex-ante* approaches to make an airline network robust and less susceptible to possible delays and disruptions. The first approach examines a plan in terms of the ease of recovery in the event of a disruption, and the second approach aims to develop a plan which is more resistant to flight delays. Within the first category of *ex-ante* models, there are three streams of research: (i) dividing an airline network into isolated subnetworks, (ii) creating crew schedules in which crew follow the same aircraft, and (iii) maximizing swapping opportunities for aircraft in the event of delays. In the first stream of literature, namely isolating airline sub-networks to improve the robustness, (Kang, 2004) consider a degradable airline schedule, which is defined as a schedule divided into several independent layers, with each layer at a different level of importance, and the layer with more importance is recovered first after a disruption. Their method improves robustness of the airline network significantly because the disruption in one layer does not affect other layers. (Rosenberger et al., 2004) considers both hub isolation and short cancellation cycles in a fleet assignment model. They demonstrate that their approach can limit the effect of a disruption to only a few flights within one particular hub.

In the second stream of literature, namely creating crew schedules so that crew follow the same aircraft, (Ehrgott and Ryan, 2002) propose a bi-criteria optimization framework to get a Pareto optimal solution by minimizing the crew pairing cost and the penalty cost incurred by allowing the

crew to change aircraft when the connection time is not long enough to absorb the expected arrival delay. (Mercier et al., 2005) try to include a penalty cost on a restricted connection in an integrated aircraft routing and crew pairing problem. Further, (Weide et al., 2010) consider the integrated problem by solving the subproblem iteratively until there is no more opportunity to improve the robustness.

In the third stream of literature, namely maximizing aircraft swapping opportunities, (Ageeva, 2000) attempts to incorporate robustness into the aircraft routing problem by increasing aircraft swapping opportunities to explore the tradeoff between robustness and schedule planning cost. (Smith and Johnson, 2006) develop a fleet assignment model by imposing station purity, limiting the number of fleet types allowed to serve each airport. They demonstrate that imposing station purity can provide more swapping opportunities for the aircraft. Further, (Gao et al., 2009) extend the station purity idea to both fleet purity and crew purity. In our research, we choose to combine the aircrafts with same seating capacity into one type, which can help increase the station purity before we optimize the assignment between incoming and outgoing flights.

In the second approach to build robustness, namely constructing a plan which is less susceptible to the possible delays, there are two methods. The first method utilizes the robust optimization to minimize the maximal possible total propagated delay while building flight schedules or aircraft routes. The second method tries to use the stochastic optimization to minimize the expectation of the total propagated delay while building flight schedules or aircraft routes. In the stream of research using the robust optimization approach, (Yan and Kung, 2016) try to minimize the maximal possible total propagated delay in the aircraft routing problem when the flight leg delays lie in a pre-specified uncertainty set. On the other hand, (Antunes et al., 2019) apply the robust optimization approach to develop a robust crew pairing schedule. Their approach can help capture the detailed delay propagation through crew connections and cost structure of crew salary. (Marla et al., 2018) compare the stochastic optimization approach, robust optimization approach and the chance-constrained optimization approach, where chance-constrained optimization approach allows the constraint violations up to a certain specified probability limit. They show that the stochastic optimization approach can improve on-time performance, total propagated delay and passenger disruptions significantly, and solutions obtained by the two other approaches can improve these criteria as well if they are formulated properly.

In the stream of research that uses the stochastic optimization approach, there are also two different methods. The first method tries to decrease the delay propagation by retiming the flight schedule for fixed aircraft assignment, and the second method intends to reduce the delay propagation by optimizing the aircraft assignment for a fixed flight schedule. Within the category of research trying to retime the flight schedule for a fixed aircraft assignment, (Lan et al., 2006) try to minimize passenger disruptions by retiming the departure times of flights within a small time window. (Ahmadbeygi et al., 2010) apply propagation tree to reduce the delay propagation by modifying the departure time of the flight so that the slack time present in the network can be re-allocated to where it is needed most. Further, (Dunbar et al., 2014) consider retiming the flight departure times to minimize the delay propagation between the aircraft and crew.

For the category of research trying to optimize the aircraft assignment for a fixed flight schedule, (Lan et al., 2006) try to reduce delay propagation by routing aircraft intelligently. In order to capture the delay propagation along each string, they apply a log-normal distribution to approximate the primary delay distribution. Here, the primary delay is the delay incurred with reasons independent of the aircraft assignment. They show that their method can greatly reduce the delay propagation. (Dunbar et al., 2012) try to minimize the total delay propagation in a combined routing and crewing network by assuming flight leg delay is known prior to the assignment of aircraft and crew to flights. (Borndörfer et al., 2010) present an alternative approach to formulate the aircraft routing problem by minimizing the total probability of positive delay propagation along an aircraft route. They demonstrate that their approach can help reduce the delay propagation by analyzing the real-world data.

Our research lies in the category of research trying to construct a plan that is less susceptible to potential delays and disruptions by optimizing the aircraft assignment for a given (fixed) flight schedule. Papers by (Lan et al., 2006), (Borndörfer et al., 2010) and (Dunbar et al., 2012) study this problem using a string-based approach by identifying the strings that minimizes the delay propagation. The main drawback in their approach is that it leads to an integer program that is not efficiently solvable for a real-world network. In reality, a major airline may run a schedule with hundreds of flights flown by one aircraft type over dozens of airports in one day. The number of strings to cover such a network is exponentially large, and solving an integer program with that many strings is impractical due to the prohibitive amount of computational resources it needs. In

contrast, we choose to minimize the delay by considering a balanced assignment problem between incoming and outgoing flights of a single airline at each airport in the network for a given aircraft type. This leg-based approach decomposes a large problem into manageable assignment problems that can be easily solved using off-shelf standard algorithms. This allows us to solve aircraft routing problems with a network with thousands of flights per day.

The second feature that separates our research from the earlier research is the treatment of the random nature of the delays. (Lan et al., 2006) and (Borndörfer et al., 2010) assume that the primary delays for all flights have the same distribution. (Dunbar et al., 2012) and (Dunbar et al., 2014) refine this approach by considering the primary delays as consisting of different components whose distribution may be dependent on the time of the day. However, they test their model using a single parametric primary delay distribution for all flights for each instance, although they do vary the parametric distribution from instance to instance. (Ahmadbeygi et al., 2010) do consider the dependence of the primary delay on the upstream primary delay as well as the originating airport, and then use clustering to identify the applicable primary delay distribution. However, they aim to minimize the delay propagation by retiming flight schedule instead of optimizing aircraft assignment problem. In contrast, we consider the arrival delay of flights as an exogenous quantity driven by several covariates in the first project. We consider several explanatory covariates for the arrival delay, such as scheduled arrival time, originating airport and aircraft type. We use a data-driven approach to estimate the arrival delay distribution. Specifically, we use training and tuning steps to identify ideal clusters using the clustering method. These optimal clusters produce an empirical arrival delay distribution, which is further used to estimate the assignment cost. In the second project, we propose a data-driven approach to estimate the primary delay distribution by considering each flight as the covariate of primary delay. Specifically, we clustering the primary delays and identify the ideal number of clusters in the training step, and then test its performance on a new set. Even though (Yan and Kung, 2016) also use a data-driven approach to deal with the primary delay distribution, they try to solve the problem by using robust optimization approach through minimizing the maximal propagated delay. By contrast, we solve the problem by using stochastic optimization approach through minimizing the total expected propagated delay, and we use a completely different data-driven approach.

Finally, in the field of stochastic assignment problem in Operations Research, both (Aldous, 1992) and (Krokhmal and Pardalos, 2009) analyze this topic by assuming elements of the assignment cost matrix are i.i.d random variables with a known parametric distribution, such as uniform distribution and exponential distribution. They then show the lower bound, upper bound and the limit of the expected cost. Further, (Emami et al., 2018) try to find the optimal assignment of jobs (measurements) to workers (tracks) in the field of multi-target tracking, so that the total assignment cost is minimized. They assume elements of the cost matrix are i.i.d random variables with a known distribution as well. In contrast, elements in our cost matrix are not i.i.d since the same arrival delay distribution of a flight determines the assignment cost of connecting that flight to all outgoing flights. Also, elements in our cost matrix have a special structure where assignment costs are always decreasing in a column (scheduled departure time). Estimating the cost matrix with this special structure is non-trivial. Hence, we propose a data-driven approach to estimate the stochastic cost matrix of the assignment problem by considering covariates such as scheduled arrival time, originating airport, etc.

CHAPTER 4

Data-Driven Aircraft Assignment at A Single Airport to Minimize Delay Propagation

4.1 Modeling the Aircraft Assignment Problem

An airline operates many flights in a day across its network. Legacy carriers such as American, Delta, United and Southwest operate thousands of flights each day. Every flight has a scheduled/actual departure time and a scheduled/actual arrival time, an originating airport and a destination airport. We will next illustrate the aircraft routing problem with a simple example.

Flight DL673 operated by Delta on July 24th, 2018 originated from Newark Airport (EWR) with destination as Atlanta Airport (ATL). It had a scheduled departure time of 2:34 pm, but actually departed at 2:40 pm from EWR, and was scheduled to arrive at ATL at 4:59 pm on the same day but actually arrived at 5:13 pm at ATL resulting in a 14 minutes' arrival delay. The aircraft that was assigned to that flight was a Boeing 717-200, with seating capacity of 100, and with a tail number N607AT. Based on the aircraft type and the airport, it is estimated to take 40 minutes to service the aircraft once it lands at ATL, so it is estimated to be ready to fly on the next flight at 5:39 pm. Thus, this aircraft tail number, N607AT, can be potentially assigned to any Boeing 717-200 flight scheduled to fly out after 5:39 pm from ATL. It was actually assigned to flight DL1779, which was scheduled to leave ATL at 6:40 pm resulting in a ground buffer of 61 minutes at ATL airport. Since the arrival delay of 14 minutes was smaller than the ground buffer of 61 minutes, the late arrival of the incoming flight did not result in any propagated delay. However, if this tail number had been assigned to another flight departing at 5:39 pm, the 14 minutes' arrival delay would have propagated to the departing flight. In general, we are interested in the problem of assigning incoming aircraft tail numbers to outgoing flights at a given airport in order to minimize propagated delay. We shall make this problem more precise below.

We consider all the flights at a single airport that are currently using aircraft of a given type (e.g., Boeing 717-200). We consider n incoming flights and n outgoing flights of a single aircraft type at a single airport over a finite time. Here, we use the term "arrival i" to represent incoming flight i, and "departure j" to denote outgoing flight j ($1 \leq i, j \leq n$). Let a_i be the scheduled arrival time for flight i ($1 \leq i \leq n$). Let d_j be the scheduled departure time for flight j ($1 \leq j \leq n$). Without loss of generality, we assume $0 \leq a_1 \leq a_2 ... \leq a_n$ and $0 \leq d_1 \leq d_2 ... \leq d_n$. Also, let A_i be the actual arrival time for flight i, and let $X_i = A_i - a_i$ be the arrival delay for incoming flight i, which may be positive or negative. Let τ be the minimum time needed to service the aircraft so that it is ready to depart on the next flight, called the minimum turnaround time. We define ready time $r_i = A_i + \tau$, which is the time when the aircraft arriving as flight i is ready to depart on its next flight.

We say that arrival i is assigned to departure j if the aircraft carrying passengers on incoming flight i is assigned to carry passengers on outgoing flight j. Define the signed propagated departure delay of flight j resulting from assigning the incoming aircraft from arrival i to departure j as

$$S_{ij} = A_i + \tau - d_j.$$

We call S_{ij} as the signed delay since it can be positive or negative. Let C_{ij} be the random "cost" of assigning arrival i to departure j. We assume

$$C_{ij} = f(S_{ij})$$

for a given function f. For example, we may use

$$f(S_{ij}) = \max(S_{ij}, 0)$$

if the "cost" of assigning incoming flight i to outgoing flight j is the non-negative propagated delay.

Let

$$x_{ij} = \begin{cases} 1, & \text{if arrival } i \text{ is assigned to departure } j, \\ 0, & \text{otherwise.} \end{cases}$$

Let $x = [x_{ij}]$ be a $n \times n$ matrix. We say x represents an assignment policy if it satisfies the following constraints:

$$\begin{aligned} &\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n, \\ &\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ &\qquad x_{ij} = 0 \text{ or } 1, \quad 1 \le i, j \le n. \end{aligned} \tag{4.1}$$

The total random cost of assignment x is given by

$$C(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

The expected cost of assignment x is

$$c(x) = E(C(x)) = E\left(\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij} \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} E(C_{ij}) x_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}, \tag{4.2}$$

where $c_{ij} = E(C_{ij})$.

We aim to find an assignment x that minimizes $c(x)$. That is, we solve

$$\begin{aligned} \textbf{SAP}: \ \min_{x} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n, \\ & \sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ & \qquad x_{ij} = 0 \text{ or } 1, \quad 1 \le i, j \le n. \end{aligned} \tag{4.3}$$

Note that problem **SAP** is a stochastic version of the traditional assignment problem since the assignment costs c_{ij} depends on the probabilistic distribution of the random arrival time, A_i, of flight i. A key challenge in solving this problem is estimating the costs c_{ij} from historical data, since future arrival delays are unknown during the planning phase. Once c_{ij} is estimated from available data, problem **SAP** can be easily solved using standard algorithms for the assignment

problem. We first focus on deriving insight on the structure of the solution to problem **SAP** in section 4.1, and then use this structure to provide a solution to problem **SAP** in section 4.2

The assignment x^* giving the minimum cost is called the optimal assignment. We state a useful result about the optimal solution x^* when the cost matrix c has a special structure. We begin with a definition:

Definition 4.1. A n by n matrix $c = [c_{ij}]$ is called a Monge matrix if

$$c_{ij} + c_{mk} \leq c_{ik} + c_{mj}$$

for all $1 \leq i < m \leq n$ and $1 \leq j < k \leq n$.

(Burkard et al., 1996) state (see Theorem 3.1 in Burkard et al. 1996) that the optimal solution x^* to **SAP** is given by the identity assignment

$$x^*_{i,i} = 1, \quad 1 \leq i \leq n,$$

if the cost matrix is a Monge matrix. And the Monge property is recently addressed by (Estes and Ball, 2021) as well. Since the a_i and d_i's are increasing in i, we see that the identity assignment is a First-In-First-Out (FIFO) assignment. We shall consider several special instances of the c matrix in the next section.

4.2 The Optimal Assignment under Deterministic Arrival Times

In this section, we consider the special case when arrival times are deterministic, that is, $A_i = a_i$ for $i = 1, ..., n$. This case is called deterministic case. Also, it implies $S_{ij} = s_{ij} = a_i + \tau - d_j = r_i - d_j$ $(1 \leq i, j \leq n)$. We study the optimal assignment under different situations below.

4.2.1 Nonnegative Delay

In this subsection, we consider the case of (non-negative) delay, that is

$$c_{ij} = \max\{s_{ij}, 0\} = s^+_{ij}, \quad 1 \leq i, j \leq n. \tag{4.4}$$

It implies the aircraft can only depart on or after the scheduled departure time. We show the special structure of the c matrix defined above in the theorem below.

Theorem 1. *The c matrix defined by Equation 4.4 is a Monge matrix.*

Proof. Let i, j, k, m satisfy $1 \leq i < m \leq n$ and $1 \leq j < k \leq n$. We need to prove

$$c_{ij} + c_{mk} \leq c_{mj} + c_{ik}.$$

We have

$$c_{ij} + c_{mk} = \max\{s_{ij}, 0\} + \max\{s_{mk}, 0\},$$

and

$$c_{mj} + c_{ik} = \max\{s_{mj}, 0\} + \max\{s_{ik}, 0\}.$$

We use the fact that c_{ij} increases in i and decreases in j and prove the result by considering the following six exclusive and exhaustive cases.

(i) If $s_{ij} \leq 0$, $s_{mk} \leq 0$, $s_{mj} \geq 0$ and $s_{ik} \leq 0$, then

$$c_{ij} + c_{mk} = 0 \leq c_{mj} + c_{ik} = c_{mj}.$$

(ii) If $s_{ij} \geq 0$, $s_{mk} \leq 0$, $s_{mj} \geq 0$ and $s_{ik} \leq 0$, then

$$c_{ij} + c_{mk} = s_{ij} \leq s_{mj} = c_{mj} + c_{ik}.$$

(iii) If $s_{ij} \geq 0$, $s_{mk} \geq 0$, $s_{mj} \geq 0$ and $s_{ik} \leq 0$, then

$$c_{ij} + c_{mk} = s_{ij} + s_{mk} \leq s_{mj} + s_{ik} \leq s_{mj} = c_{mj} + c_{ik}.$$

(iv) If $s_{ij} \leq 0$, $s_{mk} \geq 0$, $s_{mj} \geq 0$ and $s_{ik} \leq 0$, then

$$c_{ij} + c_{mk} = s_{mk} \leq s_{mj} = c_{mj} + c_{ik}.$$

(v) If $s_{ij} \leq 0$, $s_{mk} \leq 0$, $s_{mj} \leq 0$ and $s_{ik} \leq 0$, then

$$c_{ij} + c_{mk} = 0 = c_{mj} + c_{ik}.$$

(vi) If $s_{ij} \geq 0$, $s_{mk} \geq 0$, $s_{mj} \geq 0$ and $s_{ik} \geq 0$, then

$$c_{ij} + c_{mk} = s_{ij} + s_{mk} \leq s_{mj} + s_{ik} = c_{mj} + c_{ik}.$$

The result now follows. □

This immediately yields the next result:

Theorem 2. *Under the cost structure of Equation 4.4, the FIFO assignment is an optimal policy.*

Proof. Since c is a Monge matrix, the result follows from Theorem 3.1 of (Burkard et al., 1996).

□

4.2.2 Signed Delay

Next, we consider the signed delay given by

$$c_{ij} = s_{ij} = a_i + \tau - d_j = r_i - d_j, \quad 1 \leq i, j \leq n. \tag{4.5}$$

Since a_i is increasing in i and d_j is increasing in j, c_{ij} increases in i and decreases in j $(1 \leq i, j \leq n)$.

Lemma 1. *The signed delay matrix $c = [c_{ij}]$ is a Monge matrix.*

Proof. Let i, j, k, m satisfy $1 \leq i < m \leq n$ and $1 \leq j < k \leq n$. We have

$$c_{ij} + c_{mk} = r_i - d_j + r_m - d_k = r_m - d_j + r_i - d_k = c_{mj} + c_{ik}.$$

Thus, s is a Monge matrix. □

As stated before, Theorem 3.1 in (Burkard et al., 1996) implies that the FIFO assignment is optimal. However, in this case, the s matrix is even more structured than being a Monge matrix. Hence, we can obtain further results about optimal assignments in this case. The first result appears in Theorem 3 below.

Theorem 3. *Under the cost structure given in Equation 4.5, all assignments are optimal with the total signed delay given by*

$$\sum_{i=1}^{n} a_i - \sum_{j=1}^{n} d_j + n\tau.$$

Proof. We have

$$\begin{aligned} c(x) &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(a_i + \tau - d_j)x_{ij} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i x_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n}\tau x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} d_j x_{ij} \\ &= \sum_{i=1}^{n} a_i - \sum_{j=1}^{n} d_j + n\tau, \end{aligned} \tag{4.6}$$

using Equations 5.1. This proves the theorem. □

Since all the assignments are the same in this case, we consider the majorization criteria. We need to introduce the following notation. For any vector $z \in R^n$, let

$$z_{[1]} \geq ... \geq z_{[n]}$$

denote its components in decreasing order, and let

$$z_{[]} = [z_{[1]}, ..., z_{[n]}]$$

be the z vector with components arranged in decreasing order.

Definition 4.2. (Majorization) A vector $z \in R^n$ is majorized by $y \in R^n$ (or y majorizes z), if

$$\sum_{i=1}^{r} z_{[i]} \leq \sum_{i=1}^{r} y_{[i]}, \quad 1 \leq r < n, \quad \sum_{i=1}^{n} z_{[i]} = \sum_{i=1}^{n} y_{[i]}.$$

For a given assignment x, we define $a(x)$ as a vector whose j-th component is given by

$$a_j(x) = \sum_{i=1}^{n} c_{ij}x_{ij}, \quad 1 \leq j \leq n.$$

We say that an assignment x majorizes assignment y if $a(x)$ majorizes $a(y)$.

We aim to find an assignment x which can be majorized by any other assignment. In other words, we want to find an assignment x that minimizes

$$c_r(x) = \sum_{j=1}^{r} a_{[j]}(x)$$

for every $r = 1, ..., n$. Note that such an assignment is not guaranteed to exist.

That is, for a given r $(1 \leq r \leq n)$, we want to solve

$$\begin{aligned} \min_{x} \quad & c_r(x) = \sum_{j=1}^{r} a_{[j]}(x) \\ \text{s.t.} \quad & \sum_{i=1}^{n} c_{ij} x_{ij} = a(x)_j, \quad 1 \leq j \leq n \\ & \sum_{i=1}^{n} x_{ij} = 1, \quad 1 \leq j \leq n \\ & \sum_{j=1}^{n} x_{ij} = 1, \quad 1 \leq i \leq n \\ & x_{ij} = 0 \text{ or } 1, \quad 1 \leq i, j \leq n \end{aligned} \tag{4.7}$$

We show the surprising result that the FIFO assignment optimizes the above objective function for each r.

Theorem 4. *Under the cost structure of Equation 4.5, the FIFO assignment is majorized by all other assignments.*

Proof. Let assignment x be the FIFO assignment policy, that is, $x_{ii} = 1$ for all i $(1 \leq i \leq n)$. Let y be another assignment. We shall show that

$$c_r(x) \leq c_r(y)$$

for all $1 \leq r \leq n$.

If $y = x$, we are done. If y is not equal to x, then there is an i for which $x_{ii} = 1$, but $y_{ii} = 0$. Let i be the smallest such index. Then there exists a $k > i$ such that y_{ik}=1 and an $m > i$ such

that $y_{mi} = 1$. Now construct an assignment y' by setting $y'_{ii} = 1$, $y'_{mk} = 1$, and the rest of y' is the same as y.

Due to the monotonicity properties of the c matrix, we need to consider two cases. We first consider the case when $c_{ik} < c_{mk} < c_{ii} < c_{mi}$. Let $a_{[]}(y')$ and $a_{[]}(y)$ be the reordered vectors. Then $a_{[]}(y) = [..., c_{mi}, ..., c_{ik}, ...]$ and $a_{[]}(y') = [..., c_{ii}, ..., c_{mk}, ...]$, elements represented by ... are the same in both the vectors. Since $c_{ii} < c_{mi}$, and $c_{ii} + c_{mk} = c_{ik} + c_{mi}$, so $a(y')$ is majorized by $a(y)$. Similar results hold in the second case when $c_{ik} < c_{ii} < c_{mk} < c_{ki}$. Thus, y' is majorized by y. That is

$$c_r(y') \leq c_r(y)$$

for all $1 \leq r \leq n$. If $y' = x$, we are done, else we repeat. Note that y' is strictly "closer" to x. Since there is a finite number of assignments, and we never see the same assignment more than once, this procedure will terminate with $y' = x$. The result follows, since y was arbitrary.

□

Based on the definition of majorization criterion, we see that Theorem 4 can help illustrate the robustness of the FIFO assignment in a distributional sense, i.e., the FIFO assignment not only minimizes the total delays, but also minimizes the maximum delay, the two largest delays, the three largest delays, and so on. While the FIFO assignment policy is optimal in minimizing propagated delays when flight arrival times are deterministic, the FIFO policy is not necessarily optimal when flight arrival times are stochastic as shown in the next section. However, given the strong properties of the FIFO assignment in a deterministic setting, it acts as a good benchmark for evaluating any solution when arrival times are stochastic.

4.3 The Optimal Assignment under Stochastic Arrival Times

In this section, we consider the real-world setting where the actual arrival times A_is are unknown during the planning phase and, hence, are random variables. This setting is called the stochastic case. The expected cost of assigning arriving flight i to departing flight j is thus given by

$$c_{ij} = E(\max\{A_i + \tau - d_j, 0\}) = E(\max\{a_i + X_i + \tau - d_j, 0\}), \quad 1 \leq i, j \leq n. \tag{4.8}$$

4.3.1 Example

We use one simple example to show that the optimal assignment derived under the stochastic case can perform better than the FIFO assignment. Suppose there are 2 incoming flights and 2 outgoing flights at a given airport, that is $n = 2$. Let $a_1 = 0.3$, $a_2 = 0.5$, $\tau = 0.7$, $d_1 = 1.3$, and $d_2 = 1.5$. Suppose the delay for the i-th flight is $X_i \sim N(\mu_i, \sigma_i^2)$ $(1 \leq i \leq n)$. Let h_i and H_i be the pdf and cdf, respectively, of X_i. Let $\mu_1 = 0.4$, $\mu_2 = 0$. We also have the variances of the delay as $\sigma_1^2 = \sigma_2^2 = 0.04$.

For the expected nonnegative delay, we compute the cost matrix using Equation 4.8. Here, we use the following formula for a random variable X with $N(\mu, \sigma^2)$ distribution :

$$E(max(X, \alpha)) = \mu(1 - H(\alpha)) + \sigma^2 h(\alpha) - \alpha(1 - H(\alpha)),$$

where α is a constant, and h and H are pdf and cdf of X, respectively. We get the following cost matrix for the 2×2 assignment for this example

$$c = \begin{bmatrix} 0.1396 & 0.0396 \\ 0.0396 & 0.0059 \end{bmatrix},$$

with optimal assignment

$$x^* = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The total expected non-negative delay of the FIFO assignment policy is $.1396 + .0059 = .1455$. However, the optimal assignment x^* has the total expected non-negative delay $.0396 + .0396 = .0792$. Thus, the optimal policy based on the stochastic analysis performs strictly better than the FIFO assignment. Hence, in this section we focus on analyzing solutions to the assignment problem for the stochastic case.

4.3.2 The Revised FIFO Assignment

In this subsection, we create a modified version of the FIFO assignment for the stochastic case, where we do not solve the assignment problem in the stochastic case directly. Instead, we define a

new cost matrix c' as follows:

$$c'_{ij} = \max\{a_i + E(X_i) + \tau - d_j, 0\} = \max\{a_i + \mu_i + \tau - d_j, 0\} = \max\{a'_i + \tau - d_j, 0\}, \quad 1 \le i, j \le n, \tag{4.9}$$

where $\mu_i = E(X_i)$ is the expected delay of flight i, and $a'_i = a_i + \mu_i$, which can be called the expected arrival time. Let

$$c_{ij}(X_i) = \max\{a_i + X_i + \tau - d_j, 0\}, \quad 1 \le i, j \le n.$$

Based on Jensen's inequality, we know

$$c_{ij}(E(X_i)) \le E(c_{ij}(X_i)) \tag{4.10}$$

since $c_{ij}(X_i)$ is a convex function of X_i $(1 \le i, j \le n)$. We can reindex the arriving flights so that $a'_i \le a'_{i+1}$, $1 \le i < n$. Based on Theorem 2, we know the FIFO assignment is an optimal policy under c' with the reindexed arrivals. We call this as the revised FIFO (written as rFIFO) assignment. Under the rFIFO assignment, the arriving flight with the i-th smallest a'_i is assigned to outgoing flight i.

Let

$$c'(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c'_{ij} x_{ij}, \tag{4.11}$$

be the total cost using assignment x under the cost structure of Equation 4.9. Let x'^* be the optimal assignment under this cost structure. The cost $c(x)$ is defined in Equation 4.2, which is the total cost applying assignment x under the cost structure of Equation 4.8. Let x^* be the optimal assignment under this cost structure. Because of the special relationship between c'_{ij} and c_{ij} in Equation 4.10, we show that there is a lower bound and an upper bound on $c(x^*)$ in Theorem 5.

Theorem 5. *Under the cost structure of Equations 4.2 and 4.11,* $c'(x'^*) \le c(x^*) \le c(x'^*)$.

Proof.

$$c'(x'^*) \le c'(x^*) \le c(x^*) \le c(x'^*).$$

Here the first inequality follows since x'^* is optimal for c', and the second inequality follows from Equation 4.10, and the last inequality follows because x^* is optimal for c.

□

From Theorem 5, we see that $c'(x'^*)$ provides a lower bound for $c(x^*)$ without knowing the cost matrix c. Furthermore, $c(x'^*)$ provides an upper bound for $c(x^*)$ without solving the assignment problem under cost matrix c.

4.3.3 The Stochastic Assignment

We now consider solving the assignment problem under the cost structure of Equation 4.8 directly. The derived optimal assignment is called the stochastic assignment. However, solving this problem may involve as many as 30,000 flights for some aircraft types when we solve the assignment problem for each aircraft type for all flights over a three month period. This can slow down the solution of the assignment problem considerably. Hence, we introduce a daily assignment algorithm that decomposes the original problem into T smaller daily assignment problems.

Let $n_a(t)$ ($n_d(t)$) be the number of scheduled arrivals (scheduled departures, resp.) on day t, $1 \leq t \leq T$. A flight is called an overnight flight for day t if it is scheduled to arrive before day t and it is not scheduled to depart on day t or later. Let $O(t)$ be the number of overnight flights on day t. Here, we let $O(1) = 0$. We see that $n_a(t) + O(t)$ is the number of flights that are available to depart on day t. Of these $n_d(t)$ are actually scheduled to depart on day t, and the rest are overnight flights for day $t + 1$. Hence, we get

$$O(t + 1) = n_a(t) + O(t) - n_d(t).$$

Clearly, $n_a(t)$ and $n_d(t)$ are known from the data, and is guaranteed to be such that $O(t) \geq 0$.

We can decompose the assignment problem into T smaller assignment problems if we assume that the $O(t)$ overnight flights are assigned to the first $O(t)$ departing flights on day t. We describe the day-by-day assignment problem below.

Let cumulative number of scheduled arrivals up to day t be defined as $n_{ca}(t) = \sum_{t'=1}^{t'=t-1} n_a(t')$ for $1 < t \leq T + 1$, with $n_{ca}(1) = 0$. Thus, the $n_a(t)$ arriving flights on day t are indexed (in increasing order of their scheduled arrival time) from $n_{ca}(t) + 1$ to $n_{ca}(t + 1)$. The t-th assignment problem

consists of assigning these flights to departing flights indexed from $n_{ca}(t)+1$ to $n_{ca}(t)+n_d(t)-O(t)$, and $O(t+1)$ overnight flights if $n_d(t)-O(t) > 0$. Otherwise, these incoming flights will be assigned to the $O(t+1)$ overnight flights directly. This gives us the entire assignment solution as a union of the T smaller assignments.

4.4 A Data-Driven Approach for the Aircraft Assignment Problem

In this section, we first illustrate a data-driven approach to solve the stochastic aircraft assignment problem, and then introduce a similar data-driven approach for the rFIFO assignment in brief.

4.4.1 Data-Driven Approach to the Stochastic Assignment

Clearly, the central component of the stochastic assignment model is the estimation of assignment cost matrix, which is derived through the estimation of the arrival delay X_i of flight i for $1 \leq i \leq n$. However, since arrival delays are unknown during the planning phase, the challenge in solving this problem arises in estimating the (stochastic) assignment costs associated with assigning the tail number of an arriving flight to a departing flight at that airport since the assignment cost depends on arrival delays. Arrival delays depend on several covariates such as originating airport, aircraft type, airport congestion, and time varying factors such as scheduled arrival time of day, day of the week, and month of the year.

We use a data-driven approach to estimate the assignment costs for our aircraft assignment problem. Specifically, we estimate the assignment costs by using empirical observations of arrival delays from prior years' flight records to compute the estimated propagated delay associated with connecting an arriving aircraft tail number to a departing flight. We propose a data-driven clustering method to account for factors such as originating airport of an arriving flight, scheduled time of arrival, and aircraft type to translate the empirical observations of prior years' arrival delays to the assignment costs for our model. These assignment costs are used to test the performance of several aircraft assignment policies using a hold-out sample data set.

We implemented several well developed procedures for this estimation: regression tree, random forest, Neural net clustering and clustering method. The total propagated delays obtained by

using the FIFO, regression tree, random forest, neural net clustering and k-means clustering (to be defined below) methods are shown in Table 4.1. From the table, we see that the k-means clustering method performs the best. Hence, we focus on the clustering method as described in detail below.

Table 4.1: The Total Actual Propagated Delay (in Minutes) by Using FIFO, Regression Tree, Random Forest, Neural Net Clustering and k-means Clustering Methods in 2018

Aircraft type	FIFO	Regression tree	Random forest	Neural net clustering	k-means clustering
A319-114	0	0	0	0	0
Boeing 757-351	2589	3143	2575	2610	2553
Boeing 737-732	743	814	530	713	854
Boeing 737-832	578	589	564	504	600
A320-200	2584	3231	2586	3218	3186
Boeing 737-932ER	7826	8402	8464	7971	7610
Boeing 757-200	8866	8716	8751	8695	7617
Boeing 717-200	33743	33545	32702	32111	28496
A321-211	35977	41565	35256	36462	34780
MD-88/MD-90-30	137405	141521	129028	137664	132840
Total	230311	241526	220456	229948	218536

We illustrate this approach using data from 2016, 2017 and 2018. To be more specific, we first cluster flights in our training data (e.g., 2016) based on covariates such as scheduled arrival time, originating airport and aircraft type using nine deciles of observed arrival delays. These clusters provide empirical data to estimate the arrival delay distribution for flights in the validation data (e.g., 2017). This is then used to estimate the assignment costs to derive the optimal stochastic assignment for the validation data (e.g., 2017). We then choose the optimal number of clusters based on the performance of this optimal assignment based on actual arrival delays in the validation data (e.g., 2017). Finally, we test the actual performance of this clustering method based optimal stochastic assignment, using the optimal number of clusters from the validation data, on a hold-out test data set (e.g., 2018).

Our approach controls for three important covariates that drive the arrival delay of any flight i: its scheduled arrival time (SA), its originating airport (OA), and the aircraft type (AT). We show the rationale on why we choose these three covariates in Appendix A. We control for the variation in arrival delays based on scheduled arrival time, the flight origin airport, and the aircraft type using a clustering procedure described in detail below.

We first describe the clustering procedure to estimate the empirical arrival delay distribution based on the scheduled arrival-time (SA) covariate. We begin by classifying each flight based on a one-hour time block in which it is scheduled to arrive. For example, a flight scheduled to arrive at 7:12 am is classified in the [7am-7:59am] time block. We thus classify each flight into one of the 24 arrival-time blocks. Since the number of arriving flights between midnight and 7 am is very small, we reduce the number of arrival time blocks to 17 by creating one homogenous midnight till 7 am time block. We then group the 17 time blocks into cl^{sa} clusters, so that the arrival delay distribution of flights within each one-hour block in a given cluster are similar. We next provide the technical description of this clustering process.

Let SA be the scheduled arrival time of a flight in number of hours past midnight. It takes values in [0,24). Then we discretize it by splitting $[0, 24)$ into 17 subintervals: $I_1 = [0, 7), I_2 = [7, 8), I_3 = [8, 9), \cdots, I_{16} = [21, 22), I_{17} = [22, 24)$. The number of flights in each subinterval in our dataset is shown in Table 4.2.

Table 4.2: Number of Flights for Each Time of Day Subinterval

Subintervals	[0,7)	[7,8)	[8,9)	[9,10)	[10,11)	[11,12)	[12,13)	[13,14)	[14,15)	[15,16)	[16,17)	[17,18)	[18,19)	[19,20)	[20,21)	[21,22)	[22,24)
Number of flights	1894	3596	6358	4041	3017	3858	3052	3281	4419	4615	3733	2349	4265	2887	5038	3812	1725

Let

$$g(SA) = k, \quad \text{if } SA \in I_k.$$

Thus, the continuous variable SA is converted into a categorical variable $g(SA)$ taking integer values from 1 through 17. Now let

$$F_{SA}(k) = \{i : 1 \leq i \leq n, g(SA_i) = k\}, \quad 1 \leq k \leq 17,$$

be the set of flights with arrival time in interval I_k. We compute $\delta_m(k)$, the $10m$-th percentile $(1 \leq m \leq 9)$ of the empirical distribution of the observed delay of flights in $F_{SA}(k)$ using the data from 2016. Let $\delta(k) = [\delta_1(k), \delta_2(k), \cdots, \delta_9(k)]$ be a vector of these nine deciles for a given k. Thus, the n flights yield the following 17 data points

$$(k, \delta(k)), \quad 1 \leq k \leq 17.$$

We use k-means clustering method to cluster δ into cl^{sa} clusters, where cl^{sa} is a given integer. k-means clustering method assumes that the number of clusters cl^{sa} is a given integer. It aims to minimize the sum of the within-cluster variances (squared Euclidean distances). The resulting clustering has the property that the data points in the same cluster have similar arrival delay distribution. Thus, the clustering algorithm produces a cluster function

$$clsa : \{1, 2, \cdots, 17\} \to \{1, 2, \cdots, cl^{sa}\}.$$

Under this cluster function, the i-th flight belongs to cluster $CLA_i = clsa(g(SA_i))$ based on its scheduled arrival time.

Next, we consider originating airport (OA) as the covariate under consideration, which is already a categorical variable. There are many originating airports, but some of them only have a few flights between the originating airport and the hub airport in the data set that we analyze. We put all airports with less than or equal to 250 flights into one cluster. For other airports with more than 250 flights, we use the clustering method described below for further analysis. Let n_{oa} be the number of originating airports with more than 250 flights in our data. For these originating airports, the number of flights is shown in Table 4.3.

Table 4.3: Number of Flights for Each Originating Airport

Airport	Number of flights	Airport	Number of flights	Airport	Number of flights	Airport	Number of flights	Airport	Number of flights
ABQ	269	DCA	1112	IND	798	MYR	318	SAV	924
ALB	253	DEN	746	JAN	629	OKC	511	SDF	737
AUS	622	DFW	998	JAX	1184	OMA	375	SEA	662
BDL	586	DSM	262	JFK	551	ORD	998	SFO	569
BHM	893	DTW	993	LAS	641	ORF	768	SJU	361
BNA	962	ECP	524	LAX	723	PBI	947	SLC	651
BOS	969	EWR	955	LGA	1363	PDX	348	SNA	265
BUF	431	EYW	303	LIT	562	PHL	926	SRQ	509
BWI	991	FLL	1237	MCI	752	PHX	488	STL	689
CAE	292	FNT	258	MCO	1458	PIT	738	SYR	264
CAK	338	GPT	276	MDT	251	PNS	626	TLH	358
CHS	950	GRR	354	MDW	678	PVD	261	TPA	1297
CLE	671	GSO	522	MEM	874	PWM	265	TYS	285
CLT	985	GSP	684	MIA	1076	RDU	1053	VPS	429
CMH	730	HOU	398	MKE	636	RIC	756	other	4204
CVG	602	HSV	568	MLB	342	ROC	264		
DAB	399	IAD	578	MSN	259	RSW	765		
DAL	419	IAH	665	MSP	951	SAN	413		
DAY	349	ICT	266	MSY	1179	SAT	599		

The goal of our clustering procedure is to group origin airports into clusters such that airports belonging to the same cluster have similar arrival delay distribution. A technical description of the clustering procedure based on origin airport is provided next. Following the same procedure as above, we let

$$F_{OA}(k) = \{i : 1 \leq i \leq n, OA_i = k\}, \quad 1 \leq k \leq n_{oa},$$

be the set of flights with originating airport k. We compute, the $10m$-th percentile ($1 \leq m \leq 9$) of the empirical distribution of the observed delay of flights in $F_{OA}(k)$ using the data from 2016. Let $\eta(k) = [\eta_1(k), \eta_2(k), \cdots, \eta_9(k)]$ be a vector of these nine deciles for a given airport k. Thus, the n flights give us the following n_{oa} data points

$$(k, \eta(k)), \quad 1 \leq k \leq n_{oa}.$$

We use the same k-means clustering method to cluster η into cl^{oa} clusters, where cl^{oa} is a given integer. Thus, we have the following cluster function for OA.

$$cloa : \{1, 2, \cdots, n_{oa}\} \to \{1, 2, \cdots, cl^{oa}\}.$$

Under this cluster function, the i-th flight belongs to cluster $CLO_i = cloa(OA_i)$.

Finally, we consider aircraft type (AT) as the covariate under consideration, which is already a categorical variable. Let n_{at} be the number of aircraft types in our data set. For these aircraft type, the number of flights is shown in Table 4.4. The goal of our clustering procedure is to group aircraft type into clusters such that aircraft type belonging to the same cluster have a similar arrival delay distribution. A technical description of the clustering procedure based on aircraft type is provided next. Following the same procedure as above, we let

$$F_{AT}(k) = \{i : 1 \leq i \leq n, AT_i = k\}, \quad 1 \leq k \leq n_{at},$$

be the set of flights with aircraft type k. We compute, the $10m$-th percentile ($1 \leq m \leq 10$) of the empirical distribution of the observed delay of flights in $F_{AT}(k)$ using the data from 2016. Let $\gamma(k) = [\gamma_1(k), \gamma_2(k), \cdots, \gamma_9(k)]$ be a vector of these nine deciles for a given aircraft type k. Thus,

the n flights give us the following n_{at} data points

$$(k, \gamma(k)), \quad 1 \leq k \leq n_{at}.$$

We use the same k-means clustering method to cluster γ into cl^{at} clusters, where cl^{at} is a given integer. Thus, we have the following cluster function for AT.

$$clat : \{1, 2, \cdots, n_{at}\} \rightarrow \{1, 2, \cdots, cl^{at}\}.$$

Under this cluster function, the i-th flight belongs to cluster $CLAT_i = clat(AT_i)$.

Finally, we combine clusters based on flight scheduled arrival time, originating airport and aircraft type to create joint clusters as described below. For $u \in \{1, 2, \cdots, cl^{sa}\}$, $v \in \{1, 2, \cdots, cl^{oa}\}$, $w \in \{1, 2, \cdots, cl^{at}\}$, we define

$$F'(u, v, w) = \{i : 1 \leq i \leq n, CLA_i = u,\ CLO_i = v,\ CLAT_i = w\}.$$

Let n_{uvw} be the cardinality of $F'(u, v, w)$. Thus $F'(u, v, w)$ describes a joint cluster through combining the originating airport clusters, scheduled arrival time clusters, and aircraft type clusters.

We next use these joint clusters $F'(u, v, w)$ to estimate the assignment costs for our stochastic aircraft assignment problem. For flights in our validation or test data sets (e.g., 2017 or 2018), we use the arrival delay data and clusters from the training data set (e.g., 2016) to estimate the assignment costs. For example, to estimate the assignment cost of connecting arriving flight i to departing flight j in 2017, we first check the cluster to which flight i belongs based on its scheduled arrival time, originating airport and aircraft type, and then use the empirical arrival delay distribution of this cluster in 2016 to approximate its assignment cost in 2017. We next provide a detailed mathematical description of this process.

The estimated cost of assigning arriving flight $i \in F'(u, v, w)$ to departing flight j using cl^{sa} clusters of SA, cl^{oa} clusters of OA, and cl^{at} clusters of AT in year yr is given by

$$\begin{aligned} c_{ij}(cl^{sa}, cl^{oa}, cl^{at}, yr) &= E(\max\{a_{i,yr} + \tau_{yr} + X_{i,yr} - d_{j,yr}, 0\}) && (4.12) \\ &\approx \frac{1}{n_{uvw}} \Sigma_{k \in F'(u,v,w)} \max\{a_{i,yr} + \tau_{2016} + x_{k,2016} - d_{j,yr}, 0\}, && (4.13) \end{aligned}$$

where $x_{k,2016}$ is the observed arrival delay of flight k in 2016, and $a_{i,yr}$ and $d_{j,yr}$ are from year yr, $yr = 2017$ or 2018. Also, τ_{yr} is the minimum turnaround time in year yr for a given aircraft type at a given airport.

The above process for estimating assignment costs can be conducted for any arbitrary number of clusters cl^{sa}, cl^{oa}, and cl^{at}. Obviously, increasing the number of clusters will provide an increasingly better in-sample fit but may have a poor out-of-sample performance. We next describe a procedure to find the optimal number of clusters cl^{sa}, cl^{oa}, and cl^{at} so that it has the best performance on actual arrival delays from a validation data-set (e.g., $yr = 2017$).

For a given number of clusters cl^{sa}, cl^{oa}, and cl^{at}, we first solve the assignment problem for a validation data-set (e.g., $yr = 2017$).

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(cl^{sa}, cl^{oa}, cl^{at}, yr)x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n, \\ & \sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ & x_{ij} = 0 \text{ or } 1, \quad 1 \le i, j \le n. \end{aligned} \tag{4.14}$$

Let the optimal solution to the above assignment problem be denoted by $x^*(cl^{sa}, cl^{oa}, cl^{at}, yr)$ for given cl^{sa}, cl^{oa}, and cl^{at}. We further discuss how to measure the performance of $x^*(cl^{sa}, cl^{oa}, cl^{at}, yr)$ if it is implemented in year yr. In this case, we need to compute the intrinsic travel time and the propagated delay.

We first show how to compute the intrinsic travel time by using the actual data. Let $Y_{i,yr}$ be the intrinsic travel time of flight i in year yr. Suppose an airplane has flown a string of k flights in the data set. Without loss of generality, suppose the flights in that string are indexed $1-2-3-...-k$. Then we have

$$Y_{1,yr} = A_{1,yr} - d_{1,yr}, \tag{4.15}$$

$$Y_{i,yr} = A_{i,yr} - d_{i,yr} - (a_{i-1,yr} + Y_{i-1,yr} + \tau_{yr} - d_{i,yr})^{+}, \quad 2 \le i \le k, \tag{4.16}$$

where $A_{i,yr}$ is the actual observed arrival time for flight i, and $a_{i,yr}$ and $d_{i,yr}$ are the scheduled arrival and departure times for flight i in year yr, respectively. Thus, we can derive the intrinsic travel time for all flights by using the actual data. Since we have data about the flights over an extended period from the past, we can compute the intrinsic travel time of each flight i. This intrinsic travel time is independent of the assignment. Thus it can be used to evaluate the performance of any assignment.

Now we discuss how we can compute the propagated delay of any proposed assignment x at the airport we analyze assuming the assignments at other airports are the same as the actual assignment used by the airline. Note that a given assignment x together with the assignments at other airports specifies a unique string of flights for each aircraft in the flight network. Consider a single string consisting of k flights. Assume, without loss of generality, that the flights are numbered 1, 2, 3,..., k. Note that the value of the intrinsic travel time $Y_{i,yr}$ of flight i $(1 \leq i \leq k)$ is known from the data analysis as described earlier. Let $A_i(yr, x)$ be the implied arrival time of flight i in this string under assignment x in year yr. Then we have

$$A_1(yr, x) = d_{1,yr} + Y_{1,yr}, \tag{4.17}$$

$$A_i(yr, x) = d_{i,yr} + Y_{i,yr} + (A_{i-1}(yr, x) + \tau_{yr} - d_{i,yr})^+, \quad 2 \leq i \leq k. \tag{4.18}$$

Carrying out these calculations for all the strings in the flight network, we can compute the implied arrival time $A_i(yr, x)$ of each flight i. We define the actual propagated delay of assigning arrival i to departure j under assignment x in the validation year yr as

$$c^a_{ij}(yr, x) = \max\{A_i(yr, x) + \tau_{yr} - d_{j,yr}, 0\} \quad (1 \leq i, j \leq n). \tag{4.19}$$

Let

$$c^a(cl^{sa}, cl^{oa}, cl^{at}, yr) = \sum_{i=1}^{n} \sum_{j=1}^{n} c^a_{ij}(yr, x^*(cl^{sa}, cl^{oa}, cl^{at}, yr)) x^*_{ij}(cl^{sa}, cl^{oa}, cl^{at}, yr),$$

be the total actual propagated delay using assignment $x^*(cl^{sa}, cl^{oa}, cl^{at}, yr)$. Thus, $c^a(cl^{sa}, cl^{oa}, cl^{at}, yr)$ measures the actual total propagated delay that would have resulted if the assignment costs es-

timated from the training data (e.g., 2016) were used to create an optimal assignment for flights in the validation data (e.g., 2017) and implemented in the validation data based on the cluster parameters cl^{sa}, cl^{oa}, and cl^{at}. We now tune our model to find the optimal number of clusters cl^{sa*}, cl^{oa*}, and cl^{at*} as follows:

$$(cl^{sa*}, cl^{oa*}, cl^{at*}) = \text{argmin}\{c^a(cl^{sa}, cl^{oa}, cl^{at}, 2017) : 1 \leq cl^{sa} \leq 7, 1 \leq cl^{oa} \leq 9, 1 \leq cl^{at} \leq 4\}. \tag{4.20}$$

There may be several pairs of $(cl^{sa}, cl^{oa}, cl^{at})$ that have the same minimum total actual propagated delay. We choose the one that has smallest cl^{sa} first. If there are several such pairs, we choose the one with smallest cl^{oa}. If there are still several such pairs, we further choose the one with smallest cl^{at}.

Finally, we test the performance of this method on a hold-out test data set (e.g., 2018). Specifically, we use the optimal number of clusters derived in the previous step, cl^{sa*}, cl^{oa*}, and cl^{at*} to estimate the assignment costs for flights in the test data set using the empirical arrival delay distribution from the training data. We then solve the stochastic assignment problem using these estimated assignment costs to derive the optimal aircraft assignment for flights in the test data. We then calculate the total actual propagated delay in the test data set, $c^a(cl^{sa*}, cl^{oa*}, cl^{at*}, 2018)$, using Equations 4.14 and 4.19, to compute the performance of our proposed method.

4.4.2 Data-Driven Approach to the rFIFO Assignment

In this subsection, we describe a data-driven approach to derive a revised version of the FIFO assignment, called rFIFO, when arrival delays are stochastic. Compared to the data-driven approach to derive the stochastic assignment, we replace Equation 4.12 with Equation 4.21 to derive the estimated cost of assigning arrival $i \in F'(u, v, w)$ to departure j using cl^{sa} clusters of SA, cl^{oa} clusters of OA, and cl^{at} clusters of AT in year yr. That is,

$$\begin{aligned} c_{ij,r}(cl^{sa}, cl^{oa}, cl^{at}, yr) &= \max\{a_{i,yr} + \tau_{yr} + EX_{i,yr} - d_{j,yr}, 0\} && (4.21) \\ &\approx \max\{a_{i,yr} + \tau_{2016} + \frac{\Sigma_{k \in F'(u,v,w)} x_{k,2016}}{n_{uvw}} - d_{j,yr}, 0\}. && (4.22) \end{aligned}$$

We then find the optimal number of clusters cl^{sa}, cl^{oa} and cl^{at} based on performance on flights in the validation data, i.e., $yr = 2017$. Namely, we first solve Equation 4.14 with cost matrix $c_r(cl^{sa}, cl^{oa}, cl^{at}, yr) = [c_{ij,r}(cl^{sa}, cl^{oa}, cl^{at}, yr)]$. Suppose the optimal assignment by using this updated cost matrix is $x^*_r(cl^{sa}, cl^{oa}, cl^{at}, yr)$. Let

$$c^a_r(cl^{sa}, cl^{oa}, cl^{at}, yr) = \sum_{i=1}^{n}\sum_{j=1}^{n} c^a_{ij}(yr, x^*_{ij,r}(cl^{sa}, cl^{oa}, cl^{at}, yr))x^*_{ij,r}(cl^{sa}, cl^{oa}, cl^{at}, yr),$$

which is the total actual propagated delay using assignment $x^*_r(cl^{sa}, cl^{oa}, cl^{at}, yr)$. Then we can derive the optimal number of clusters as follows

$$(cl^{sa*}_r, cl^{oa*}_r, cl^{at*}_r) = \text{argmin}\{c^a_r(cl^{sa}, cl^{oa}, cl^{at}, 2017) : 1 \le cl^{sa} \le 7, 1 \le cl^{oa} \le 9, 1 \le cl^{at} \le 4\}. \tag{4.23}$$

There may be several pairs of $(cl^{sa}, cl^{oa}, cl^{at})$ that have the same total actual propagated delay as well. We use the same rule to find the optimal number of clusters as in the previous subsection. Finally, we use the optimal clusters, cl^{sa*}_r, cl^{oa*}_r, and cl^{at*}_r, to derive the optimal rFIFO assignment in the test data from 2018, and its corresponding total actual propagated delay $c^a_r(cl^{sa*}_r, cl^{oa*}_r, cl^{at*}_r, 2018)$.

4.5 Computational Experiments

In this section, we first describe how we collect and clean the data to illustrate our analysis using training, validation, and test data sets. We then describe our data-driven approach to estimate the assignment costs using the training data set (2016) and then derive the optimal number of clusters for the rFIFO and stochastic assignments using the validation data set (2017). Using these assignment costs and optimal clusters, we then solve the stochastic aircraft assignment problem for the test data set (2018). We compare the total actual propoagated delay performance of three policies (FIFO, rFIFO, and stochastic assignment) in the test data set (2018) to develop our recommendations for the aircraft assignment problem.

4.5.1 Data Collection and Cleaning

We collected the data from several resources within the Bureau of Transportation Statistics (BTS) website. The main data is the Airline On-time Performance data for all domestic flights flown in the US. This data set includes actual/scheduled arrival/departure times, arrival/departure delays, originating airport, destination airport and the tail number of the aircraft for each commercial flight operated by major airlines in the US. From this website, we downloaded the data for the months of July, August and September in 2016, 2017 and 2018. Thus, the number of days of our analysis is 92 for each year. There are 4,809,651 flight records in this downloaded data set. For the purpose of illustration, we chose to focus on Delta Airlines, thus reducing the size of the data set to 740,191 flight records, among which 276 flights do not have tail number. Also, we downloaded Aircraft Registration Master file, Deregistered Aircraft file and Aircraft Reference file from Aircraft Registry Database to get the aircraft type and seating capacity for each aircraft. In addition, some tail numbers were invalid/missing in the Aircraft registration file. Hence, we conducted an internet search to find the aircraft type and seating capacity for tail numbers with missing information in the Aircraft registration file. Then, using the tools from (Ramdas and Williams, 2006) and (Arıkan et al., 2013), we cleaned the data. Specifically, we first deleted flights with inaccurate information, such as the flights with an invalid tail number, flights with more than 684 minutes of air-time, flights with negative taxi-in time, taxi-out time or air time, and flight with a ratio of distance/air-time higher than 10.45. Further, to avoid corrupt data, we eliminated flights with following conditions (i) the actual departure time of an aircraft from an airport is earlier than its actual arrival of the previous flight to the same airport, (ii) duplicate records for the same aircraft flying the same route on succesive flights, (iii) an aircraft arrives to an airport and the next flight of the same aircraft is from a different airport in less than 5 hours.

We then chose Atlanta airport, the world's busiest airport, for Delta Airlines to conduct our analysis. There are eleven aircraft types flown by Delta Airlines at Atlanta airport. However, we removed the data for Boeing 767 from our analysis because Delta was phasing out this aircraft type during the period of our analysis. The following detailed information for each aircraft type is shown in Table 4.4:

1. the type of aircraft,

2. seating capacity,

3. $n(yr)$ = the number of incoming/outgoing flights for each aircraft type in year yr (yr =2016, 2017 and 2018),

The rows in the table are arranged in the increasing order of the number of flights in 2018. They range from about 500 to almost 30,000 over the 92 days. Next, we consider the originating airports for the flights arriving at Atlanta. There are 90 originating airports with more than 250 flights, shown in Table 4.3. We do not distinguish between the smaller airports.

Table 4.4: Number of Flights by Aircraft Type in Each Year

Aircraft type	Seating capacity	n(2016)	n(2017)	n(2018)
A319-114	145	1341	364	551
Boeing 757-351	275	1155	674	658
Boeing 737-732	149	1280	1128	1493
Boeing 737-832	189	1368	2856	1652
A320-200	182	1914	2808	2938
Boeing 737-932ER	222	5314	4878	5694
Boeing 757-200	178	6207	5497	6240
Boeing 717-200	100	10502	7404	7211
A321-211	199	1721	5070	7325
MD-88/MD-90-30	142	30611	29718	28628

4.5.2 Optimal Number of Clusters

To implement the data-driven approach, we need to derive the optimal number of clusters (based on the arrival times, the originating airports and the aircraft types) under the rFIFO and stochastic assignments using the total actual propagated delay in 2017.

For the stochastic assignment, the optimal number of clusters derived based on the procedure in section 4.4.1 for aircraft type is 1, i.e., we regard all the aircraft types as a single homogenous cluster. After fixing the optimal number of clusters for the aircraft type, the total actual propagated delay in 2017 under the stochastic assignment as a function of cl^{sa} and cl^{oa} is shown in Table 4.5.

From Table 4.5, we can see the optimal numbers of clusters for the scheduled arrival time and originating airport are both 5. The optimal clusters for scheduled arrival time under the stochastic

Table 4.5: Total Actual Propagated Delay (in Minutes) Under the Stochastic Assignment in 2017

cl^{sa} \ cl^{oa}	1	2	3	4	5	6	7	8	9
1	208403	197878	201461	201798	204293	204281	200297	202491	200827
2	206673	200443	199516	204920	202254	199520	206225	203033	203672
3	204009	201745	203177	201227	197992	200241	201211	204652	206162
4	205199	199369	198275	202578	202049	207285	198093	198360	201649
5	209272	202380	204606	202373	**193713**	201026	202775	200196	202964
6	203451	204933	203312	202618	198802	199851	202778	206720	197453
7	204877	200708	200454	196380	198999	198263	206276	200508	201116

assignment are shown in Table 4.6. From the table, we can see that the scheduled arrival time is split into five intervals, namely, $[0, 14)$, $[14, 17)$, $[17, 20)$, $[20, 22)$ and $[22, 24)$. Specifically, $[17, 20)$, and $[20, 22)$ are the evening peak times, and $[0, 14)$, $[14, 17)$ and $[22, 24)$ are the rest. Further, the optimal clusters for originating airport under the stochastic assignment are shown in Table 4.7. We see that the airports in clusters 2, 4 and 5 are mostly busy airports. The airports in cluster 3 are mostly small airports and not busy, while the airports in cluster 1 are in between. From these results, we can see that our data-driven approach under the stochastic assignment makes intuitive sense.

Table 4.6: Optimal Clusters for the Scheduled Arrival Time Under the rFIFO and Stochastic Assignments

Cluster	Interval
1	[0,14)
2	[14,17)
3	[17,20)
4	[20,22)
5	[22,24)

For the rFIFO assignment, the optimal number of clusters for the aircraft type is 3. The corresponding optimal clusters are shown in Table 4.8. From the table, we can see that aircraft types in cluster 1 include Boeing 737-732, Boeing 737-832 and MD-88/MD-90-30, where MD-88/MD-90-30 has the largest number of flights flown at ATL, and the other two types may be overwhelmed by MD-88/MD-90-30. Aircraft types in cluster 2 including A319-114, Boeing 737-

Table 4.7: Optimal Clusters for the Originating Airport Under the Stochastic Assignment

Cluster	Airport
1	BDL, BOS, CLT, ICT, JAX, LIT, PBI, PHX, PIT, PVD, PWM, RIC, ROC, RSW, SAV, SDF, SNA, SYR, TPA
2	BHM, BNA, BUF, CLE, CMH, DAB, DAY, DEN, DTW, FLL, GSP, JAN, LAS, LAX, MCI, MDT, MDW, MIA, MSN, MSP, MSY, ORF, SAN, SAT, SEA, SFO, SLC, SRQ, STL
3	ABQ, ALB, CAE, CAK, CHS, CVG, DSM, ECP, FNT, GPT, GRR, GSO, HSV, MKE, MYR, OKC, OMA, PDX, PNS, SJU, TLH, TYS, VPS
4	AUS, BWI, DAL, DFW, EYW, HOU, IAD, IAH, IND, MCO, MEM, MLB, RDU
5	DCA, EWR, JFK, LGA, ORD, PHL

932ER and Boeing 757-200 mostly have a small number of flights flown at ATL. And aircraft types in cluster 3 including Boeing 757-351, A320-200, A321-211 and Boeing 717-200 are in between.

Table 4.8: Optimal Clusters for the Aircraft Type Under the rFIFO Assignment

Cluster	Aircraft type
1	Boeing 737-732, Boeing 737-832, MD-88/MD-90-30
2	A319-114, Boeing 737-932ER, Boeing 757-200
3	Boeing 757-351, A320-200, A321-211, Boeing 717-200

After fixing the optimal number of clusters for the aircraft type, the total actual propagated delay in 2017 under the rFIFO assignment with the change of cl^{sa} and cl^{oa} is shown in Table 4.9.

Table 4.9: Total Actual Propagated Delay (in Minutes) Under the rFIFO Assignment in 2017

cl^{sa} \ cl^{oa}	1	2	3	4	5	6	7	8	9
1	202783	200105	201422	198407	200102	200323	197897	198717	199843
2	203699	204224	201599	200306	198912	201129	199811	199310	197957
3	204347	203709	202844	201762	200688	201279	200769	200498	199241
4	204696	203729	200772	202845	199323	197274	198925	201952	201273
5	204701	203850	200394	202920	196998	**195755**	196417	199515	198841
6	204836	203570	201396	202489	201245	199308	197984	201355	197880
7	204340	203543	200912	203252	201529	199373	197615	200571	198154

From the table, we can see that the optimal number of clusters for the scheduled arrival time is 5, which is the same as that in the stochastic assignment. And the optimal number of clusters for the originating airport is 6. The optimal clusters for originating airport under the rFIFO assignment are shown in Table 4.10. We see that the airports in clusters 2, 4 and 5 are mostly busy airports. The airports in clusters 3 and 6 are mostly small airports and not busy, while the airports

in cluster 1 are in between. This is similar to that in the stochastic case. From these results, we see that our data-driven approach under the rFIFO assignment makes intuitive sense as well. The optimal number of clusters for the scheduled arrival time, the originating airport and the aircraft type under the rFIFO and stochastic assignments are summarized in Table 4.11.

Table 4.10: Optimal Clusters for the Originating Airport Under the rFIFO Assignment

Cluster	Airport
1	BDL, BOS, CLT, CMH, ICT, JAX, LAS, LIT, MIA, ORF, PBI, PHX, PIT, PVD, PWM, RIC, ROC, RSW, SAN, SAV, SDF, SNA, SYR, TPA
2	AUS, BHM, BNA, CLE, DAB, DAY, DEN, DTW, FLL, GSP, JAN, LAX, MCI, MDT, MDW, MSN, MSP, MSY, SAT, SEA, SFO, SLC, SRQ, STL
3	ABQ, BUF, CAE, CHS, CVG, ECP, GPT, GRR, GSO, HSV, MKE, MYR, OKC, OMA, PDX, PNS, TLH, VPS
4	BWI, DAL, DFW, EYW, HOU, IAD, IAH, IND, MCO, MEM, MLB, RDU
5	DCA, EWR, JFK, LGA, ORD, PHL
6	ALB, CAK, DSM, FNT, SJU, TYS

Table 4.11: The Optimal Number of Clusters cl_r^{sa*}, cl_r^{oa*}, cl_r^{at*}, cl^{sa*}, cl^{oa*} and cl^{at*}

cl_r^{sa*}	cl_r^{oa*}	cl_r^{at*}	cl^{sa*}	cl^{oa*}	cl^{at*}
5	6	3	5	5	1

4.5.3 Comparison of FIFO, rFIFO, and Stochastic assignment policies

Using the optimal clusters derived from 2017 data as described in the previous sub-section, we first compute the assignment costs for the rFIFO and stochastic assignment policies. The optimal rFIFO and stochastic assignment policies were computed in our test data set from 2018 using these assignment costs. We then compare the performance of FIFO, rFIFO and stochastic assignments as shown in Table 4.12. In this table, $c^a(F, 2018)$ (second column) is the total actual propagated delay by using the FIFO assignment policy in 2018, while $c_r^a(cl_r^{sa*}, cl_r^{oa*}, 2018)$ (column 3) and $c^a(cl^{sa*}, cl^{oa*}, 2018)$ (column 5) represent the actual propagated delay by using the rFIFO and stochastic assignment policies, respectively. $I_{rF}(2018)$ and $I_{st}(2018)$ (columns 4 and 6) are the percentage improvement by using the rFIFO and stochastic assignments, respectively, compared to the FIFO assignment in 2018. The last row (columns 2, 3 and 5) of Table 4.12 shows the total actual propagated delay over all flights of all aircraft types in 2018 under the three assignments. We see from this table that both the rFIFO and stochastic assignments perform better than the FIFO assignment. Specifically, the stochastic assignment performs the best, with the overall improvement

compared to the FIFO assignment being 5.11%, even though the stochastic assignment may perform worse than the FIFO assignment for some individual aircraft types.

Table 4.12: Comparison Among FIFO, rFIFO and Stochastic Assignments in terms of Total Actual Propagated Delay (in Minutes) in 2018

Aircraft type	$c^a(F, 2018)$	$c_r^a(cl_r^{sa*}, cl_r^{oa*}, 2018)$	$I_{rF}(2018)$	$c^a(cl^{sa*}, cl^{oa*}, 2018)$	$I_{st}(2018)$
A319-114	0	0	0.00%	0	0.00%
Boeing 757-351	2589	2566	0.89%	2553	1.39%
Boeing 737-732	743	743	0.00%	854	-14.94%
Boeing 737-832	578	578	0.00%	600	-3.81%
A320-200	2584	2683	-3.83%	3186	-23.30%
Boeing 737-932ER	7826	7852	-0.33%	7610	2.76%
Boeing 757-200	8866	8840	0.29%	7617	14.09%
Boeing 717-200	33743	32475	3.76%	28496	15.55%
A321-211	35977	35408	1.58%	34780	3.33%
MD-88/MD-90-30	137405	135157	1.64%	132840	3.32%
Sum of all types	230311	226302	1.74%	218536	5.11%

Then we show the network effect of the rFIFO and stochastic assignments. We compare the total actual propagated delay over the entire network for different assignment policies in Table 4.13. We see that the rFIFO and stochastic assignments still perform better than the FIFO assignment. Specifically, the rFIFO assignment performs 0.68% better than the FIFO assignment, and the stochastic assignment performs 1.65% better than the FIFO assignment. If we compare the saving in the total actual propagated delay at Atlanta airport with that in the entire network in comparison to the FIFO assignment, we see that the saving increases from 4,009 $(230,311 - 226,302)$ minutes to 7,299 $(1,066,316 - 1,059,017)$ minutes under the rFIFO assignment, and the saving increases from 11,775 $(230,311 - 218,536)$ minutes to 17,590 $(1,066,316 - 1,048,726)$ minutes under the stochastic assignment. It implies that even though we only aim to minimize the propagated delay at Atlanta airport, the derived assignment can bring a further reduction in total actual propagated delay over the entire network.

We further compare the FIFO, rFIFO and stochastic assignments in 2018 in terms of the percentage of flights delayed due to propagated delay as shown in Table 5.16. For a given assignment policy x, define the percentage of flights delayed by more than b minutes in 2018 as $pd(x, b)$. Assignment x can be FIFO, rFIFO or stochastic assignment. From Table 5.16, we see that the

Table 4.13: Comparison Among FIFO, rFIFO and Stochastic Assignments in terms of Total Actual Propagated Delay (in Minutes) in the Network in 2018

Aircraft type	FIFO	rFIFO	Stochatic assignment
A319-114	83908	83908	83908
Boeing 757-351	14714	14691	14675
Boeing 737-732	5711	5752	5918
Boeing 737-832	69298	69298	68373
A320-200	97379	97478	98405
Boeing 737-932ER	67487	67546	68453
Boeing 757-200	78280	78054	76107
Boeing 717-200	213405	211701	207092
A321-211	104904	101920	103241
MD-88/MD-90-30	331230	328669	322554
Sum of all types	1066316	1059017	1048726
Improvement compared to FIFO		0.68%	1.65%

difference of $pd(x, b)$ among these three assignments is very small. It implies the rFIFO or stochastic assignment can lower total actual propagated delay without significantly impacting the percentage of flights delayed as compared to the FIFO assignment.

In view of the above evidence, we recommend the stochastic assignment for implementation. To see its effectiveness, we also compared the stochastic assignment policy with the actual aircraft assignment implemented by the airline in 2018 in terms of the total actual propagated delay and the percentage of delayed flights, as shown in Tables 4.15 and 4.16, respectively. In Table 4.15, $c^a(a, 2018)$ (column 2) is the total actual propagated delay induced by using the actual airline assignment, while $c^a(cl^{sa*}, cl^{oa*}, 2018)$ (column 3) is the total actual propagated delay induced by using the stochastic assignment proposed in this project. Similarly, $I_s^a(2018)$ is the percentage improvement in terms of the total actual propagated delay by using the stochastic assignment policy over the actual airline assignment. We see that the stochastic assignment outperforms the actual assignment for all types except for types Boeing 757-351 and MD-88/MD-90-30. The reason why MD-88/MD-90-30 under actual assignment can perform better is because we observe that the airline changes the assignment of MD-88/MD-90-30 dynamically based on the updated information it has. Table 4.16 offers similar comparison in the fraction of delayed flights. From this table, we

Table 4.14: Percentage of Flights Delayed (Due to Propagated Delay) Under the FIFO, rFIFO and Stochastic Assignments

Aircraft type	FIFO		rFIFO		Stochastic assignment	
	$pd(x,0)$	$pd(x,15)$	$pd(x,0)$	$pd(x,15)$	$pd(x,0)$	$pd(x,15)$
A319-114	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Boeing 757-351	8.97%	5.62%	8.21%	5.02%	8.21%	4.86%
Boeing 737-732	0.47%	0.40%	0.40%	0.33%	0.40%	0.27%
Boeing 737-832	0.18%	0.18%	0.18%	0.18%	0.24%	0.24%
A320-200	0.92%	0.68%	1.06%	0.82%	1.09%	0.92%
Boeing 737-932ER	2.27%	1.42%	2.34%	1.49%	2.32%	1.60%
Boeing 757-200	1.04%	0.95%	1.01%	0.85%	1.01%	0.83%
Boeing 717-200	3.52%	3.02%	3.37%	2.80%	3.36%	2.75%
A321-211	9.31%	5.75%	9.88%	5.79%	10.83%	5.90%
MD-88/MD-90-30	5.01%	4.00%	4.92%	3.80%	5.18%	3.90%
All types	4.26%	3.19%	4.27%	3.07%	4.50%	3.14%

see that the stochastic assignment brings a significant improvement in fraction of delayed flights compared to the actual assignment for each aircraft type.

The last row in Tables 4.15 and 4.16 shows the impact of using stochastic assignment policy for all flights. For example, it shows a large 18% improvement in the total actual propagated delay over the actual assignment used in 2018, from 265,419 minutes under the actual assignment to 218,536 minutes under the stochastic assignment. The percentage of flights delayed by over 15 minutes decreases from 6.89% in actual assignment to 3.14% under the stochastic assignment policy. This implies that if the airline can use the stochastic assignment, it can not only greatly reduce the delay propagation, but also improve the on-time performance across all flights significantly.

Further, to derive insight on how the policies differ in their assignment from the FIFO policy, we derive the number of flights whose assignment differs from the FIFO assignment in Table 4.17. We see that the number of flight assignments that differ from the FIFO assignment gradually increases from rFIFO to stochastic to the actual airline assignment. Specifically, the actual airline assignment is almost 50% different from the FIFO assignment. This suggests that the airline can achieve significant performance improvement by moving closer to the FIFO assignment as suggested by the stochastic assignment policy. This table also suggests that the stochastic assignment differs from the FIFO assignment for a small number of flights for aircraft types with small number

Table 4.15: Comparison Between the Actual Assignment and Stochastic Assignment on the Total Actual Propagated Delay (in Minutes)

Aircraft type	Actual assignment	Stochastic assignment	Improvement
	$c^a(a, 2018)$	$c^a(cl^{sa*}, cl^{oa*}, 2018)$	$I_s^a(2018)$
A319-114	1251	0	100.00%
Boeing 757-351	2070	2553	-23.33%
Boeing 737-732	4935	854	82.70%
Boeing 737-832	4559	600	86.84%
A320-200	19971	3186	84.05%
Boeing 737-932ER	12488	7610	39.06%
Boeing 757-200	20317	7617	62.51%
Boeing 717-200	32842	28496	13.23%
A321-211	36129	34780	3.73%
MD-88/MD-90-30	130857	132840	-1.52%
All types	265419	218536	17.66%

Table 4.16: Comparison Between the Actual Assignment and Stochastic Assignment on the Percentage of Delayed Flights (Due to Propagated Delay)

Aircraft type	Actual assignment		Stochastic assignment	
	$pd(x, 0)$	$pd(x, 15)$	$pd(x, 0)$	$pd(x, 15)$
A319-114	6.17%	4.17%	0.00%	0.00%
Boeing 757-351	10.94%	6.69%	8.21%	4.86%
Boeing 737-732	9.71%	5.29%	0.40%	0.27%
Boeing 737-832	8.41%	4.36%	0.24%	0.24%
A320-200	15.28%	9.36%	1.09%	0.92%
Boeing 737-932ER	9.15%	4.23%	2.32%	1.60%
Boeing 757-200	12.48%	5.90%	1.01%	0.83%
Boeing 717-200	12.18%	6.42%	3.36%	2.75%
A321-211	17.72%	8.38%	10.83%	5.90%
MD-88/MD-90-30	14.17%	7.41%	5.18%	3.90%
All types	13.42%	6.89%	4.50%	3.14%

of flights. However, for aircraft types with large number of flights, such as MD-88/MD-90, the stochastic assignment policy also differs significantly from the FIFO policy. Thus, the FIFO policy could be significantly sub-optimal when travel times are stochastic.

Table 4.17: Number of Flights that Differ from FIFO Assignment for Different Aircraft-assignment Policies

Aircraft type	$n(2018)$	rFIFO	Stochastic assignment	Actual assignment
A319-114	551	32	108	215
Boeing 757-351	658	20	67	217
Boeing 737-732	1493	39	416	668
Boeing 737-832	1652	26	464	767
A320-200	2938	132	1040	1406
Boeing 737-932ER	5694	410	2104	2762
Boeing 757-200	6240	336	2576	3125
Boeing 717-200	7211	869	3091	3550
A321-211	7325	772	2543	3166
MD-88/MD-90-30	28628	8533	13372	14201

Finally, to illustrate the benefit of our proposed model, we calculate the potential monetary savings for an airline if they were to implement the stochastic assignment policy. We first convert total flight delays into a dollar amount by using the per minute delay cost estimate given by the Airlines for America (A4A). The cost per minute for the different impacts of delays is shown in Table 4.18 (Airlines for America 2019).

Table 4.18: Airlines for America Per Minute Delay Cost Estimate

Item	Delay cost for Airlines ($/min.)
Fuel	27.02
Crew-Pilots/Flight Attendants	23.36
Maintenance	11.75
Aircraft Ownership	9.28
Other	2.80

Note that this estimate does not consider the cost to the passenger, or the environment including noise and emissions issues (Barnhart et al. 2014 and Chen and Solak 2015), but only includes the cost incurred by an airline due to flight delays. We use the direct costs related to crew-pilots/flights attendants, aircraft ownership, and other costs to compute the per minute propagated delay cost,

which is $35.44 in 2018. Thus, based on the propagated delay in Table 4.15, we can work out the savings in cost by using our stochastic assignment policy compared to the actual assignment for all aircraft types. This gives us a total savings of $(265419 - 218536) * 35.44 = 1.66$ million dollars for the three months of July, August, and September. Thus, the corresponding annual savings is around $1.66 * 243710/62390 = 6.49$ million dollars, based on 243710 total number of incoming flights operated by Delta Airlines at Atlanta airport in 2018, while 62390 is the corresponding total number of incoming flights from July to September in 2018. It implies that it helps Delta Airlines potentially save approximately 6.5 million dollars if they had used the stochastic assignment at Atlanta airport in 2018. Thus, an aircraft assignment policy that takes into account the stochastic nature of propagated delays can potentially reduce operating costs related to flight delays significantly.

4.6 Maintenance Routing Problem

In this section, we analyze the maintenance routing problem by considering the maintenance checks mandated by FAA. There are four types of maintenance checks: A, B, C, and D, varying in scope, duration, and frequency (Clarke et al. 1997 and Lan et al. 2006). Any violations may result in significant penalties (Eltoukhy et al. 2017). In literature, the research on the maintenance routing problem only considers A checks, which are currently called the line maintenance checks based on the definition from National Aviation Academy. The line maintenance checks are the only checks that need to be performed frequently since they only need to cover some basic inspection checks. The academy recommends that aircraft needs the line maintenance checks every 24 to 60 hours of accumulated flight time, but it depends on the operator of the aircraft. Since the maintenance requires trained professionals and equipment, these checks are only performed at a limited number of airports. For example, Delta Airlines has 30 maintenance stations shown in Table 4.19.

Table 4.19: Maintenance Stations for Delta Airlines

Maintenance stations	ATL, BDL, BOS, BWI, CHS, CVG, DCA, DEN, DTW, EWR, FLL, HNL, JFK, LAS, LAX, LGA, MCO, MEM, MIA, MSP, PDX, PHL, PHX, RDU, SAN, SAV, SEA, SFO, SLC, TPA

We now consider the maintenance routing problem using the stochastic assignment at Atlanta airport proposed in Section 7, and the actual assignment at other airports. This yields a path followed by each tail number. Each path is a sequence of flights flown by one tail number. We

can partition each path into a set of subpaths, namely strings. Each string starts and ends at a maintenance station with a layover larger than or equal to five hours. Each string can be further subdivided into rotations, and each rotation has a layover larger than or equal to five hours at its endpoints. Thus the entire network can be divided into rotations, with each rotation consisting of a collection of flights. We use the rotations as given from now on, and we then construct the strings based on the connection between rotations. Let L_j be the length of string j, i.e., the sum of the flight times of all the flights in the string. Then the excess time of string j is given by

$$c_j = \max(L_j - T_r, 0),$$

where T_r is the maximum flight time between maintenance stations allowed by FAA. We use $T_r = 60$ and 72 hours in our numerical experiment. Our objective is to select the strings so that the sum of excess times of the selected strings is minimized. We formulate the network below.

We describe the network formulation for the maintenance routing problem using stochastic assignment at Atlanta airport, and actual assignment at other airports. Suppose there are N airports in total, indexed 1 through N. Let $\mathbf{N} = \{1, 2, ..., N\}$. We reserve index 1 for Atlanta airport. Let there be R rotations in total, indexed 1 through R. Let $\mathbf{R} = \{1, ..., R\}$. Let $I_n \subseteq \mathbf{R}$ be the set of rotations that end in airport n, and $O_n \subseteq \mathbf{R}$ be the set of rotations leaving airport n. Let $s(r)$ be the start time of rotation r and $t(r)$ be the termination time of rotation r, $r \in \mathbf{R}$. Also, let $sa(r)$ be the starting airport of rotation r, and $ta(r)$ be the terminating airport of rotation r. Let $f(r)$ be the total flight time of rotation r.

We first construct a directed acyclic network $G = (V, E)$ with vertex set V and edge set E as follows. The vertex set is given by

$$V = \{(n, r) : n \in \mathbf{N}, \;\; r \in I_n \cup O_n\}.$$

That is, each node has a label (n, r), where $n \in \mathbf{N}$ is the airport index, and r is the rotation index for the rotation incident on airport n, i.e., $sa(r) = n$ or $ta(r) = n$. Note that there are two vertices in V for each rotation, hence the number of vertices is $|V| = 2R$.

Next we construct the edge set E and define the edge length $w(e)$ for $e \in E$. An edge e is a pair (n, r) of nodes in V. Define a partition $\mathbf{N} = \mathbf{N_1} \cup \mathbf{N_2}$ as follows: $\mathbf{N_1}$ is the set of maintenance stations, $\mathbf{N_2}$ is the set of non-maintenance stations. Then

$$\begin{aligned} E = &\{((1,r),(1,r')) : \ r \text{ is assigned to } r' \text{ based on the stochastic assignment at Atlanta airport}\} \\ &\cup \{((n,r),(n,r')) : \ r \text{ is assigned to } r' \text{ based on the actual assignment at airport } n \in \mathbf{N_1} \setminus \{1\}\} \\ &\cup \{((n,r),(n,r')) : \ n \in \mathbf{N_2},\ r \in I_n,\ r' \in O_n,\ 0 < s(r') - t(r) \leq 48 \text{ hours}\} \\ &\cup \{((n,r),(m,r)) : \ n \in \mathbf{N},\ r \in O_n,\ r \in I_m\} \end{aligned} \tag{4.24}$$

The first subset is the set of rotations connected at Atlanta airport using the stochastic assignment. The second subset describes the rotation connected in other maintenance stations. The third subset describes the possible connections of incoming rotations to outgoing rotations at non-maintenance airports, ensuring the ground time is at most 48 hours. The last subset describes a connection from airport n to airport m if rotation r starts from airport n and ends at airport m. The edges in the first three subsets have zero length, that is, $w((n,r),(n,r')) = 0$ if $((n,r),(n,r')) \in E$. And length of the edge in the last subset is the total flight time of the rotation, that is, $w((n,r),(m,r)) = f(r)$ if $((n,r),(m,r)) \in E$.

We can find all the strings for this directed acyclic network using the method proposed by (Baidari and Sajjan, 2016). Let n_s be the number of strings. The information about the strings is represented by a $R \times n_s$ matrix A, defined as follows. Let

$$A(i,j) = \begin{cases} 1, & \text{if rotation } i \text{ is contained in string } j, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathbf{c} = [c_j, j = 1, 2, ..., n_s]$ be a column vector of these excess times. Define the decision variable y_j for each string as follows.

$$y_j = \begin{cases} 1, & \text{if string } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

Let $y = [y_j, j = 1, 2, ..., n_s]$ be a column vector of these decision variables. And let $\mathbf{e}$ be a R-dimensional column vector of ones. Then a vector y is feasible if each rotation belongs to one and only one string, i.e., $A\mathbf{y} = \mathbf{e}$. The cost of this feasible assignment y is given by $\mathbf{c}^\top \mathbf{y}$. The maintenance routing problem can now be defined as identifying a feasible y with the minimum cost $\mathbf{c}^\top \mathbf{y}$. That is, the maintenance routing problem can be formulated as follows:

$$\begin{aligned} \min_{y} \quad & \mathbf{c}^\top \mathbf{y} \\ \text{s.t.} \quad & A\mathbf{y} = \mathbf{e}. \end{aligned} \tag{4.25}$$

A selection of strings can be used to create a path for each tail number over 92 days (July, August and September) we analyze. However, if we solve the problem over 92 days directly, the number of airports is 95, and the number of rotations is 13,553 for aircraft type MD-88/MD-90-30. This creates a huge network with astronomically large number of strings. And the previous research using the string-based method, such as (Lan et al., 2006) and (Yan and Kung, 2016), only consider a daily flight network by assuming the fleeted schedule will repeat everyday. By contrast, we can solve the aircraft routing problem in a two-week flight network using the method specified above. So we split the 92 days into seven two-week periods, thus we have seven subnetworks. We construct the solution over 92 days from these seven subnetworks by combining the flight times of the strings that begin in one subnetwork and end in the next. We call the resulting solution as proposed solution.

For a given solution y, we compute the following two performance measures.

$$\begin{aligned} Infs(y) &= \text{number of infeasible strings (the string with flight time exceeding } T_r) \text{ in } y. \\ Me(y) &= \text{maximum excess time of infeasible strings in } y. \end{aligned} \tag{4.26}$$

Let y^* be the proposed solution. Let y_a be the actual set of strings, and y_s be the set of strings by using the stochastic assignment at Atlanta airport, and actual assignment at other airports. We show the two performance measures, namely, $Infs(y)$ and $Me(y)$, for y_a, y_s and y^* in Tables 4.20 and 4.21, respectively. From Table 4.20, we can see that if we regard all types as a whole, the improvements of y^* compared to y_a and y_s are both larger than 30% when the required time is 60

hours, and the improvements are both larger than 40% when the required time is 72 hours. From Table 4.21, we can see y^* can help reduce the maximum excess time for the infeasible strings for six of all aircraft types compared to y_a and y_s. In particular, it can help reduce the maximum excess time for type MD-88/MD-90-30 (the type with the maximum number of flights) significantly. It implies that our proposed method can not only help reduce the number of infeasible strings, but also the maximum excess time of the infeasible strings.

Table 4.20: Comparison on the Number of Infeasible Strings

Aircraft type	60 hours			72 hours		
	y_a	y_s	y^*	y_a	y_s	y^*
A319-114	26	23	10	11	11	4
Boeing 757-351	45	34	34	6	7	7
Boeing 737-732	1	0	0	1	0	0
Boeing 737-832	70	65	59	20	18	14
A320-200	17	12	8	5	3	1
Boeing 737-932ER	49	84	49	28	35	23
Boeing 757-200	45	43	40	24	15	15
Boeing 717-200	138	124	92	69	72	41
A321-211	15	28	18	6	13	9
MD-88/MD-90-30	247	232	121	110	123	50
Sum of all types	653	645	431	280	297	164
Improvement to y_a			33.18%			44.78%
Improvement to y_s			34.00%			41.43%

4.7 Conclusions

Flight delays have a significant impact on an airline's operating cost including increased expenses for crew, fuel, and maintenance. Propagated delays due to late arriving aircraft contribute to 40% of all flight delays as reported by the Bureau of Transportation Statistics. The propagation of flight delays in an airlines' network is largely driven by factors within an airlines' control such as aircraft routing and flight scheduling decisions. The aircraft assignment problem is to assign tail numbers on scheduled arriving flights at an airport to scheduled departing flights at the same airport with the objective of minimizing propagated delays. In this project, we propose a new data-

Table 4.21: Comparison on the Maximum Excess Time (in Hours) of Infeasible Strings

Aircraft type	60 hours			72 hours		
	y_a	y_s	y^*	y_a	y_s	y^*
A319-114	81	43	35	69	31	6
Boeing 757-351	42	72	72	30	60	60
Boeing 737-732	31	0	0	19	0	0
Boeing 737-832	55	55	40	43	43	28
A320-200	56	56	27	44	44	2
Boeing 737-932ER	42	65	43	30	53	31
Boeing 757-200	61	68	68	49	56	56
Boeing 717-200	90	79	68	78	67	40
A321-211	28	29	29	16	17	17
MD-88/MD-90-30	84	81	47	72	69	38

driven approach for the aircraft assignment problem by formulating it as a balanced assignment problem between incoming and outgoing flights flown by the same aircraft type at a single airport.

We consider both deterministic and stochastic versions of the aircraft assignment problem. In the deterministic case, we prove the optimality of the First-in-First-out (FIFO) assignment policy under two different performance measures. This justifies the use of the FIFO policy as a benchmark policy due to its optimality properties. In the stochastic case, we show that the FIFO assignment policy is no longer optimal and propose the rFIFO and stochastic assignment formulations for the aircraft assignment problem.

A key challenge in solving the problem is estimating the stochastic assignment costs associated with assigning tail numbers on an arriving flight to a departing flight at the same airport. This arises because arrival delays depend on several factors such as originating airports, time of the day, aircraft type amongst others. We propose a data-driven approach to estimate the assignment costs by using empirical observations of arrival delays from prior years' flight records to compute the empirical propagated delay distribution. We propose a data-driven clustering method to account for factors such as originating airport, time of day, and aircraft type that affect the arrival delay distribution. This empirical approach is then used to compute the aircraft assignment costs which serve as an input to our stochastic assignment model.

These assignment costs are then used to derive the aircraft assignment for two policies: rFIFO and stochastic assignment policies. The rFIFO policy is a revised version of the FIFO policy which adjusts the actual arrival time of each flight by its expected delay. The stochastic assignment policy in contrast uses the empirical distribution of arrival delays in evaluating assignment costs. The assignment costs for both policies are estimated from a training data set for Delta Airlines at Atlanta airport from 2016. The optimal clusters for the empirical estimation are established using a validation data set from 2017 for Delta. Finally, the optimal rFIFO and stochastic assignment policies are derived for an out of sample data set from 2018 for Delta at Atlanta airport. We compared the rFIFO and stochastic assignment policies with the FIFO assignment, using the data-driven approach and considering the total delay, as well as the fraction of delayed flights on an out-of-sample data set from 2018.

We show that the rFIFO and stochastic assignment policies derived from the data-driven approach both perform better than the FIFO assignment, and the stochastic assignment policy performs the best in terms of total actual propagated delay. Specifically, the improvement in total propagated delay by using the stochastic assignment compared to the FIFO assignment is 5.11%. Also, the difference in the fraction of delayed flights among these three assignments is pretty small. This implies that the rFIFO or stochastic assignment lower total actual propagated delay almost without influencing the performance in the percentage of delayed flights when it is compared to the FIFO assignment.

We also compared the stochastic assignment with the actual assignment that was used by Delta in 2018. We show that for all Delta Airlines flights at Atlanta airport from July to September in 2018, the stochastic assignment policy yields a roughly 18% improvement in total actual propagated delay over the actual airline assignment. It also reduces the fraction of delayed flights from 6.89% to 3.14% based on the DOT definition of a delayed flight. Further, we estimate that Delta Airlines would have potentially saved approximately 6.5 million dollars in flight delay related operating costs if it had used the stochastic assignment in 2018. In view of this evidence, we recommend the use of the stochastic assignment policy for the aircraft assignment problem. We feel that our proposed stochastic assignment policy is specially beneficial for busy airports with a large number of flights. By incorporating the stochastic nature of arrival delays in solving the aircraft assignment problem, and combining it with a data-driven approach can potentially help an airline significantly

reduce its operating costs due to flight delays, improve passenger convenience and experience, and help the environment by reducing emissions.

In terms of the maintenance routing problem, we construct a directed acyclic network to minimize the total excess time over three months. It would bring a significant improvement in the number infeasible strings and the maximum excess time for the string. It implies that our proposed approach can also take care of the maintenance issue mandated by FAA.

CHAPTER 5

Data-Driven Aircraft Assignment Over Multiple Airports to Minimize Delay Propagation

5.1 Model Description

We consider an aircraft assignment problem to minimize the total expected propagated delay over all the flights flown by a given aircraft type (aircrafts with the same seating capacity) run by one airline. Specifically, we consider the problem from a leg-based perspective by optimizing the assignment between incoming and outgoing flights at each airport in a daily flight network. We describe it in more detail as follows.

Suppose there are P airports in the network, labeled as $1, 2, ..., P$. There are M flights flown among these airports, labeled as $1, 2, ..., M$. Let $F = \{1, 2, ..., M\}$ be the set of these flights. Let d_i/D_i be the scheduled/actual departure time at the originating airport $ori(i)$ for flight $i \in F$, and a_i/A_i be the scheduled/actual arrival time at destination airpot $des(i)$ for flight $i \in F$. At airport p, there are a set of incoming flights and a set of outgoing flights. Let $I(p)$ be the set of incoming flights at airport p, and $O(p)$ be the set of outgoing flights at airport p. However, some pairs of incoming and outgoing flights are feasible for the aircraft assignment, and some of them are not. An incoming flight $i \in I(p)$ and an outgoing flight $j \in O(p)$ is called a feasible incoming and outgoing flight pair if the outgoing flight j is supposed to depart η minutes later than the scheduled arrival time of the incoming flight i. Let $IF(p)$ be the set of incoming flights in the feasible pairs at airport p, and $OF(p)$ be the set of outgoing flights in the feasible pairs at airport p. So, $I(p) \backslash IF(p)$ is the set of infeasible incoming flights at airport p. For these flights, we add the dummy outgoing flights $OD(p)$, which can be regarded as the overnight flight to the next day. For dummy outgoing flights, we assume its scheduled departure time is a positive infinite number, and for all the dummy outgoing flights, we assume the destination airport is $P + 1$. Similarly, $O(p) \backslash OF(p)$ is the set of infeasible outgoing flights at airport p. For these flights, we add the set of dummy incoming flights

$ID(p)$, which can be regarded as the overnight flight from the previous day. For dummy incoming flights, we assume its scheduled arrival time is a negative infinite number, and for all the dummy incoming flight, we assume the originating airport is 0. Let $INC(p) = I(p) \cup ID(p)$ be the set of all incoming flights including the dummy incoming flights at airport p, and $OUT(p) = O(p) \cup OD(p)$ be the set of all outgoing flights including the dummy outgoing flights at airport p. Thus, the number of incoming flights in $INC(p)$ is the same as the number of outgoing flights in $OUT(p)$.

We further consider the assignment problem between incoming flights in $INC(p)$ and outgoing flights in $OUT(p)$ at airport p $(1 \leq p \leq P)$. Let

$$x_{ij} = \begin{cases} 1, & \text{if flight } i \in INC(p) \text{ is assigned to flight } j \in OUT(p), \\ 0, & \text{otherwise.} \end{cases}$$

Specifically, if $x_{ij} = 1$ for some $i \in ID(p)$ and $j \in O(p)$, it means the dummy incoming flight (i.e. overnight flight from the previous day) i is assigned to flight $j \in O(p)$, which implies flight j is the first flight in the string (a sequence of flights flown by one aircraft). Similarly, if $x_{ij} = 1$ for some $i \in I(p)$ and $j \in OD(p)$, it means incoming flight $i \in I(p)$ is assigned to dummy outgoing flight (i.e. overnight flight to the next day) j, which implies flight i is the last flight of the string. Let $x = [x_{ij}]$. We say x represents an assignment policy if it satisfies the following constraints:

$$\begin{aligned} \sum_{i \in INC(p)} x_{ij} &= 1, \quad j \in OUT(p),\ 1 \leq p \leq P, \\ \sum_{j \in OUT(p)} x_{ij} &= 1, \quad i \in INC(p),\ 1 \leq p \leq P, \\ x_{ij} &= 0 \text{ or } 1, \quad i \in INC(p),\ j \in OUT(p),\ 1 \leq p \leq P. \end{aligned} \tag{5.1}$$

Now we formally introduce two important concepts related to a flight: the primary delay and the propagated delay. Intuitively speaking, the propagated delay is the delay incurred by the lateness of the previous flight, which is affected by the aircraft assignment. On the other hand, the primary delay accounts for the en-route delay, passenger connection delay, and other delays that are not a function of aircraft assignment (Lan et al. 2006; Yan and Kung 2016). We show how to compute each type of delay.

Let X_i be the primary delay of flight i. Suppose an airplane has flown a string of k flights on a given day in the data set. Without loss of generality, suppose the flights in that string are indexed $1-2-3-...-k$. Then we have

$$X_1 = (A_1 - a_1)^+ \tag{5.2}$$

$$X_i = (A_i - a_i - (A_{i-1} + \tau - d_i)^+)^+, \ \ 2 \leq i \leq k. \tag{5.3}$$

Thus, we can derive the primary delay for all flights by using the actual data. Since we have data about the flights over an extended period from the past, once we can estimate the distribution of the primary delay X_i of each flight i. This distribution is independent of the assignment.

Now we discuss how we can compute the propagated delay of any proposed assignment x. Note that a given assignment x specifies a unique string of flights for each aircraft in the network. Consider a single string consisting of k flights. Assume, without loss of generality, that the flights are numbered 1, 2, ..., k. Note that the distribution of the primary delay X_i of flight i $(1 \leq i \leq k)$ is known from the data analysis as described earlier. Let $A_i(x)$ be the implied arrival time of flight i in this string under assignment x. Then we have

$$A_1(x) = a_1 + X_1, \tag{5.4}$$

$$A_i(x) = a_i + X_i + (A_{i-1}(x) + \tau - d_i)^+, \ \ 2 \leq i \leq k. \tag{5.5}$$

Carrying out such calculations for all the strings in assignment x, we can compute the implied arrival time $A_i(x)$ of each flight i $(i \in F')$. Now let flight $i \in INC(p)$ be an incoming flight and flight $j \in OUT(p)$ be an outgoing flight at airport p. Then the implied propagated delay if flight i is assigned to flight j under assignment x can be computed by

$$C_{i,j}(x) = (A_i(x) + \tau - d_j)^+. \tag{5.6}$$

Let $c_{ij}(x) = E(C_{ij}(x))$ be the expected propagated delay of assigning incoming flight $i \in INC(p)$ to outgoing flight $j \in OUT(p)$ under assignment x $(1 \leq p \leq P)$.

The total expected propagated delay of assignment x is

$$T(x) = \sum_{p=1}^{P} \sum_{i \in INC(p)} \sum_{j \in OUT(p)} c_{ij}(x) x_{ij}. \tag{5.7}$$

Thus, the aircraft assignment problem (AP) can be modeled as follows:

$$\begin{aligned} \mathbf{AP}: \min_{x} \quad & T(x) = \sum_{p=1}^{P} \sum_{i \in INC(p)} \sum_{j \in OUT(p)} c_{ij}(x) x_{ij} \\ \text{s.t.} \quad & \sum_{i \in INC(p)} x_{ij} = 1, \quad j \in OUT(p),\ 1 \le p \le P, \\ & \sum_{j \in OUT(p)} x_{ij} = 1, \quad i \in INC(p),\ 1 \le p \le P, \\ & x_{ij} = 0 \text{ or } 1, \quad i \in INC(p),\ j \in OUT(p),\ 1 \le p \le P. \end{aligned} \tag{5.8}$$

Clearly, this is a nonlinear integer programming problem since the cost matrix $c(x)$ depends on the assignment x. It is difficult to solve this problem directly. We propose an iterative algorithm to transform this nonlinear integer programming problem into a set of linear assignment problems (one assignment problem at each airport) so that it can be solved efficiently by using the traditional algorithm, namely Hungarian algorithm. We will illustrate this iterative algorithm in detail in the next section.

5.2 Iterative Algorithm

In this section, we first introduce an iterative algorithm under different cases to derive the solution (assignment) to AP. Then we show the performance of the iterative algorithm by comparing it with the approach proposed by the previous research, and finally compare the solutions derived from the algorithm under different cases.

5.2.1 Algorithm

We begin with the description of the iterative algorithm (**IA**). It is designed to stop after H iterations, where H is a preset positive integer.

Step 0: Set $h = 0$. Compute

$$\tilde{c}_{i,j}(p,0) = (a_i + \tau - d_j)^+, \;\; i \in INC(p), j \in OUT(p), \;\; 1 \le p \le P.$$

Step 1: Solve the following assignment problem for each airport $p \in \{1, 2, \cdots, P\}$:

$$\begin{aligned} \textbf{LAP}: \; \tilde{T}(p,h+1) = \min \quad & \sum_{i \in INC(p)} \sum_{j \in OUT(p)} \tilde{c}_{ij}(p,h) x_{ij} \\ \text{s.t.} \quad & \sum_{i \in INC(p)} x_{ij} = 1, \quad j \in OUT(p), \\ & \sum_{j \in OUT(p)} x_{ij} = 1, \quad i \in INC(p), \\ & x_{ij} = 0 \text{ or } 1, \quad i \in INC(p), \; j \in OUT(p). \end{aligned} \tag{5.9}$$

Denote the resulting optimal assignment at airport p by $x(p, h+1)$. Let the overall assignment be $x(h+1) = [x(p,h+1), 1 \le p \le P]$ and

$$\tilde{T}(h+1) = \sum_{p=1}^{P} \tilde{T}(p,h+1).$$

Step 2: Let $x = x(h+1)$, compute the propagated delays implied by assignment x using Equation 5.6. Set

$$\tilde{C}_{i,j}(p,h+1) = (A_i(x) + \tau - d_j)^+, \quad i \in INC(p), j \in OUT(p), \;\; 1 \le p \le P.$$

Let $\tilde{c}_{ij}(p,h+1) = E(\tilde{C}_{ij}(h+1))$, and let the overall cost matrix be $\tilde{c}(h+1) = [\tilde{c}(p,h+1), 1 \le p \le P]$.

Step 3: Set $h = h+1$. If $h \ge H$, stop. Else go to step 1.

After the algorithm terminates, we compute $T(x(h))$ for $h = 1, 2, \cdots, H$, and choose $x(h)$ that produces the smallest $T(x(h))$ as the optimal assignment, and call it x^*. Note that $\tilde{T}(h)$ is not the same as $T(x(h))$, which can be observed from Tables 5.1 , 5.2 , 5.3 and 5.4.

Intuitively, we can regard each iteration of the iterative algorithm as the operation of the flights on one day. On the initial day (iteration 0), we optimize the assignment between incoming and outgoing fights at each airport by assuming the fight will arrive as scheduled. This resulting

Table 5.1: The change of $\tilde{T}^d(h)$, $\tilde{T}^m(h)$ and $\tilde{T}^s(h)$ as Iteration Continues for N_1

Iteration	1	2	3	4	5	6	7	8	9	10
$\tilde{T}^d(h)$	0.00	10.84	10.84	10.84	10.84	10.84	10.84	10.84	10.84	10.84
$\tilde{T}^m(h)$	0.00	431.15	415.61	415.20	415.20	415.20	415.20	415.20	415.20	415.20
$\tilde{T}^s(h)$	0.00	546.39	510.06	510.06	510.06	510.06	510.06	510.06	510.06	510.06

Table 5.2: The change of $\tilde{T}^d(h)$, $\tilde{T}^m(h)$ and $\tilde{T}^s(h)$ (in minutes) as Iteration Continues for N_2

Iteration	1	2	3	4	5	6	7	8	9	10
$\tilde{T}^d(h)$	5.00	335.48	335.48	335.48	335.48	335.48	335.48	335.48	335.48	335.48
$\tilde{T}^m(h)$	5.00	754.51	753.16	753.16	753.16	753.16	753.16	753.16	753.16	753.16
$\tilde{T}^s(h)$	5.00	1067.58	1055.94	1055.55	1055.94	1055.55	1055.94	1055.55	1055.94	1055.55

Table 5.3: The change of $T^d(x(h))$, $T^m(x(h))$ and $T^s(x(h))$ (in minutes) as Iteration Continues for N_1

Iteration	1	2	3	4	5	6	7	8	9	10	min	$\frac{T(x(1))-T(x^*)}{T(x(1))} \times 100\%$
$T^d(x(h))$	**10.84**	10.84	10.84	10.84	10.84	10.84	10.84	10.84	10.84	10.84	10.84	0.00%
$T^m(x(h))$	449.55	**415.61**	415.2	415.2	415.2	415.2	415.2	415.2	415.2	415.2	415.2	7.64%
$T^s(x(h))$	569.84	510.23	**510.06**	510.06	510.06	510.06	510.06	510.06	510.06	510.06	510.06	10.49%

Table 5.4: The change of $T^d(x(h))$, $T^m(x(h))$ and $T^s(x(h))$ (in minutes) as Iteration Continues for N_2

Iteration	1	2	3	4	5	6	7	8	9	10	min	$\frac{T(x(1))-T(x^*)}{T(x(1))} \times 100\%$
$T^d(x(h))$	356.19	**335.48**	335.48	335.48	335.48	335.48	335.48	335.48	335.48	335.48	335.48	5.81%
$T^m(x(h))$	766.28	**753.16**	753.16	753.16	753.16	753.16	753.16	753.16	753.16	753.16	753.16	1.71%
$T^s(x(h))$	1078.58	1055.94	**1055.55**	1055.94	1055.55	1055.94	1055.55	1055.94	1055.55	1055.94	1055.55	2.14%

optimal assignment $x(1)$ is actually the First-In, First-Out (FIFO) assignment at each airport. This assignment is used on day 1 (iteration 1). This assignment yields a string for each aircraft, which is then used to update the implied arrival time for each flight on day 1. Using these, we compute the cost matrix $\tilde{c}(1)$ for iteration 1, which is then further used to derive a new assignment $x(2)$ for day (iteration) 2. This iterative process continues for H iterations.

Note that this is not a descent algorithm. That is, there is no guarantee that $\tilde{T}(h)$ or $T(x(h))$ will monotonically decrease as iteration continues. Hence, we terminate the algorithm after a sufficiently large number of iterations H and choose the assignment $x(h)$ that produces the smallest $T(x(h))$ for $h = 1, 2, \cdots, H$. Our numerical experience with real data indicates that the total expected propagated delay approaches a fixed value or fluctuates within a small range in the first 10 iterations, which is shown in Tables 5.3 and 5.4. Hence, we choose to stop the algorithm with $H = 10$ iterations.

Now we consider three cases of this algorithm: (1). deterministic (d), (2). mixed (m), and (3). stochastic (s). The only difference among these three cases is in the computation of $A_i(x)$. We use the superscripts d, m or s to denote the implied arrivals under each case. The explicit equations are given below.

Deterministic Case:

$$A_1^d(x) = a_1 + EX_1, \tag{5.10}$$

$$A_i^d(x) = a_i + EX_i + E(A_{i-1}^d(x) + \tau - d_i)^+, \ \ 2 \le i \le k. \tag{5.11}$$

Here, we assume that all primary delays are equal to their expected values, and hence there is no randomness in the system.

Mixed Case:

$$A_1^m(x) = a_1 + X_1, \tag{5.12}$$

$$A_i^m(x) = a_i + X_i + E(A_{i-1}^m(x) + \tau - d_i)^+, \ \ 2 \le i \le k. \tag{5.13}$$

Here, we assume that the primary delays are stochastic with known distributions. However, we replace the implied propagated delays by their expected values. This reduces the computational effort needed to compute the cost matrices in each iteration.

Stochastic Case:

$$A_1^s(x) = a_1 + X_1, \tag{5.14}$$

$$A_i^s(x) = a_i + X_i + (A_{i-1}^s(x) + \tau - d_i)^+, \;\; 2 \leq i \leq k. \tag{5.15}$$

Thus, $A_i^s(x)$ is the same as $A_i(x)$ as described in the previous section. This case takes into account the stochastic nature of the primary as well as implied propagated delays.

We denote the corresponding $C_{ij}(x)$ (equation 5.6) as $C_{ij}^d(x)$, $C_{ij}^m(x)$ and $C_{ij}^s(x)$; and the corresponding assignment produced by the algorithm as x^{*d}, x^{*m} and x^{*s}, and the corresponding $T(x)$ as $T^d(x^{*d})$, $T^m(x^{*m})$ and $T^s(x^{*s})$.

5.2.2 Performance of the Iterative Algorithm

In this subsection, we study the performance of the iterative algorithm by comparing it with the approach proposed by the previous researchers, such as (Dunbar et al., 2014) and (Yan and Kung, 2016). (Dunbar et al., 2014) and (Yan and Kung, 2016) both model the delay stochasticity by constructing a set of random scenarios Ω, where each scenario $\omega \in \Omega$ corresponds to primary delay values X^w for each flight. Then based on Equations from 5.10 to 5.15, we can derive the expected propagated delay of assigning incoming flight $i \in INC(p)$ to outgoing flight $j \in OUT(p)$ under assignment x $(1 \leq p \leq P)$ in the deterministic, mixed and stochastic cases as follows:

$$c_{ij}^d(x) = E(C_{ij}^d(x)) = \sum_{\omega \in \Omega} \frac{1}{|\Omega|} C_{ij}^{d,\omega}(x), \tag{5.16}$$

$$c_{ij}^m(x) = E(C_{ij}^m(x)) = \sum_{\omega \in \Omega} \frac{1}{|\Omega|} C_{ij}^{m,\omega}(x), \tag{5.17}$$

$$c_{ij}^s(x) = E(C_{ij}^s(x)) = \sum_{\omega \in \Omega} \frac{1}{|\Omega|} C_{ij}^{s,\omega}(x). \tag{5.18}$$

(Dunbar et al., 2014) propose two algorithms to deal with the aircraft assignment problem, namely the exact approach and the local approach. The exact approach considers all the feasible

aircraft strings $\mathbf{R}$. And for each feasible string $r \in \mathbf{R}$, they calculate its total expected propagated delay along the string based on Equation 5.18. Then they try to minimize the total expected propagated delay over all the strings in the network. However, this approach is not practical for industry-sized problems due to its lack of an efficient column generation process, which is used to find the strings with less delay. Therefore, (Yan and Kung, 2016) compare their approach with local approach proposed by (Dunbar et al., 2014). In the local approach, (Dunbar et al., 2014) calculate the total expected propagated delay along the string based on Equation 5.17.

(Yan and Kung, 2016) try to minimize the maximal possible total propagated delay in the aircraft assignment problem when the flight delays lie in a pre-specified uncertainty set. This would produce an aircraft assignment in the network, which can be further used to derive the total expected propagated delay based on Equation 5.17.

In the approaches of (Dunbar et al., 2014) and (Yan and Kung, 2016), branch-and-price solution process finds provably good solution very quickly but fails to prove optimality for a long time. The branch-and-price is a branch-and-bound method with linear programming relaxation, which is solved by using column generation at each node of the branch-and-bound tree (Lan et al. 2006). So, the runtime of branch-and-price solution process is limited to 120 seconds, which implies the total expected propagated delay derived from (Yan and Kung, 2016) and (Dunbar et al., 2014) may not be optimal.

Then we compare our iterative algorithm in the mixed case with approach proposed by (Yan and Kung, 2016) and the local approach given by (Dunbar et al., 2014). Specifically, we compare our approach with that of (Dunbar et al., 2014) and (Yan and Kung, 2016) on two of the largest aircraft types operated by one major US airline in the 31 days of August in year 2007. The characteristics of daily flight network from these two aircraft types are shown in Table 5.5. The comparison among our approach and the approaches proposed by (Dunbar et al., 2014) and (Yan and Kung, 2016) on the total expected propagated delay is shown in Table 5.6. Here, for the iterative algorithm, we set $\eta = 31$ for N_1 to make sure the corresponding assignment uses 24 aircrafts, and set $\eta = 20$ for N_2 to make sure the corresponding assignment uses 23 aircrafts. From Table 5.6, we see that the total expected propagated delay and computation time by using each approach, and the last row indicates the improvement in total expected propagated delay of our approach compared to the approach proposed by (Yan and Kung, 2016). In terms of the total expected propagated delay, our iterative

algorithm can perform 14.51% and 10.03% better than that of (Yan and Kung, 2016), respectively, in the two daily flight networks. This is a surprising level of improvement considering the algorithm in (Yan and Kung, 2016) is a heuristic optimization algorithm. In terms of the computation time, the time needed for our iterative algorithm is less than 1 second, but the approach from (Yan and Kung, 2016) and (Dunbar et al., 2014) needs hundreds of seconds. Thus, the improvement in terms of computation time is also significant.

Table 5.5: Characteristics of Two Flight Networks in (Yan and Kung, 2016)

Network	Number of flights	Number of aircraft	Minimum turnaround time (in minutes)
N_1	106	24	20
N_2	117	23	30

Table 5.6: Comparison among Different Approaches in Total Expected Propagated Delay (in minutes) and Computation Time (in seconds)

Approach	N_1		N_2	
	Delay	Computation time	Delay	Computation time
(Dunbar et al., 2014)	575.3	217.04	854.0	140.11
(Yan and Kung, 2016)	485.7	1,041.99	837.1	420.06
Iterative algorithm	415.2	0.38	753.2	0.98
Improvement	14.51%		10.03%	

5.2.3 Comparison among Deterministic, Mixed and Stochastic cases

In this subsection, we mainly compare the assignments x^{*d}, x^{*m} and x^{*s} derived from the iterative algorithm under the deterministic, mixed and stochastic cases, as well as the optimal assignments to AP under Equation 5.1 based on the cost matrices derived from Equation 5.10 to 5.15. Let $C^d_{ij}(x)$, $C^m_{ij}(x)$ and $C^s_{ij}(x)$ be as defined in Section 3.1, and let $c^d_{ij}(x)$, $c^m_{ij}(x)$ and $c^s_{ij}(x)$ be their expected values, respectively. The next lemma compares these expected values.

Lemma 2. *For any assignment x,*

$$c^d_{ij}(x) \leq c^m_{ij}(x) \leq c^s_{ij}(x). \tag{5.19}$$

Proof. Proof

For any assignment x in the network, we consider a single string consisting of k flights derived from the assignment. Assume the flights in the string are numbered 1, 2, ..., k. Then we can use the induction method to prove the lemma.

The expected propagated delay of assigning flight 1 to flight 2 under the deterministic, mixed and stochastic cases are

$$\begin{aligned} c_{12}^d(x) &= C_{12}^d(x) = (A_1^d + \tau - d_2)^+ = (a_1 + EX_1 + \tau - d_2)^+, && (5.20)\\ c_{12}^m(x) &= EC_{12}^m(x) = E(A_1^m + \tau - d_2)^+ = E(a_1 + X_1 + \tau - d_2)^+, && (5.21)\\ c_{12}^s(x) &= EC_{12}^s(x) = E(A_1^s + \tau - d_2)^+ = E(a_1 + X_1 + \tau - d_2)^+. && (5.22) \end{aligned}$$

Using Jensen's inequality, we get

$$c_{12}^d(x) \le c_{12}^m(x) = c_{12}^s(x).$$

When $i \in \{2, ..., k-1\}$, the expected propagated delay of assigning flight i to flight $i+1$ under the deterministic, mixed and stochastic cases are

$$\begin{aligned} c_{i,i+1}^d(x) &= C_{i,i+1}^d(x) = (A_i^d + \tau - d_{i+1})^+ = (a_i + EX_i + c_{i-1,i}^d(x) + \tau - d_{i+1})^+, && (5.23)\\ c_{i,i+1}^m(x) &= EC_{i,i+1}^m(x) = E(A_i^m + \tau - d_{i+1})^+ = E(a_i + X_i + c_{i-1,i}^m(x) + \tau - d_{i+1})^+, && (5.24)\\ c_{i,i+1}^s(x) &= EC_{i,i+1}^s(x) = E(A_i^s + \tau - d_{i+1})^+ = E(a_1 + X_i + C_{i-1,i}^s(x) + \tau - d_{i+1})^+. && (5.25) \end{aligned}$$

When $i = 2$, we know

$$\begin{aligned} c_{23}^d(x) &= (a_2 + EX_2 + c_{12}^d(x) + \tau - d_3)^+ \\ &\le (a_2 + EX_2 + c_{12}^m(x) + \tau - d_3)^+ \\ &\le E(a_2 + X_2 + c_{12}^m(x) + \tau - d_3)^+ \\ &= c_{23}^m(x) \qquad (5.26)\\ &= E(a_2 + X_2 + c_{12}^s(x) + \tau - d_3)^+ \\ &\le E(a_2 + X_2 + C_{12}^s(x) + \tau - d_3)^+ \\ &= c_{23}^s(x). \end{aligned}$$

Suppose for $i \in \{2, ..., k-2\}$, we have

$$c^d_{i-1,i}(x) \leq c^m_{i-1,i}(x) \leq c^s_{i-1,i}(x).$$

Then

$$\begin{aligned} c^d_{i,i+1}(x) &= (a_i + EX_i + c^d_{i-1,i}(x) + \tau - d_{i+1})^+ \\ &\leq (a_i + EX_i + c^m_{i-1,i}(x) + \tau - d_{i+1})^+ \\ &\leq E(a_i + X_i + c^m_{i-1,i}(x) + \tau - d_{i+1})^+ \\ &= c^m_{i,i+1}(x), \\ &\leq E(a_i + X_i + c^s_{i-1,i}(x) + \tau - d_{i+1})^+ \\ &\leq E(a_i + X_i + C^s_{i-1,i}(x) + \tau - d_{i+1})^+ \\ &= c^s_{i,i+1}(x). \end{aligned} \tag{5.27}$$

Thus, for $i \in \{1, ..., k-1\}$, we have

$$c^d_{i,i+1}(x) \leq c^m_{i,i+1}(x) \leq c^s_{i,i+1}(x) \tag{5.28}$$

Thus, the above inequalities hold for all the strings derived from assignment x. Hence, the Lemma follows.

□

Let x^{**d}, x^{**m} and x^{**s} be the optimal assignments to AP under Equation 5.1 based on the cost matrices $c^d(x)$, $c^m(x)$ and $c^s(x)$, respectively. These may be different from x^{*d}, x^{*m} and x^{*s} derived from the iterative algorithm since the assignments x^{*d}, x^{*m} and x^{*s} may not be the optimal assignments. Let $T^{**d} = T^d(x^{**d})$, $T^{**m} = T^m(x^{**m})$ and $T^{**s} = T^s(x^{**s})$ be the total expected propagated delay under the optimal assignments x^{**d}, x^{**m} and x^{**s}, respectively. Using Lemma 2, we get Proposition 1 below.

Proposition 1.

$$T^{**d} \leq T^{**m} \leq T^{**s}.$$

Proof. Proof We first prove $T^{**d} \leq T^{**m}$. Then $T^{**m} \leq T^{**s}$ follows similarly.

$$T^{**d} = T^d(x^{**d}) \leq T^d(x^{**m}) \leq T^m(x^{**m}) = T^{**m}.$$

Here the first inequality follows since x^{**d} is optimal for cost matrix c^d, and the second inequality follows from Lemma 2.

□

We further show the numerical comparison among $T^d(x^{*d})$, $T^m(x^{*m})$ and $T^s(x^{*s})$ on the data from the two flight networks in the previous subsection. The result is shown in Table 5.7. From the table, we see that there is a big difference among these three values for both networks. Specifically, $T^d(x^{*d})$ is much smaller than $T^m(x^{*m})$ and $T^s(x^{*s})$, which implies the approach proposed by (Dunbar et al., 2012) (only using expected primary delay) underestimates the total expected propagated delay significantly. Further, $T^m(x^{*m})$ is much smaller than $T^s(x^{*s})$, it implies the approach proposed by (Yan and Kung, 2016) and the local approach proposed by (Dunbar et al., 2014) underestimates the total expected propagated delay significantly as well. Even though (Dunbar et al., 2014) also consider the stochasticity of both primary and propagated delays, their approach is less efficient, which makes it less applicable to the industry-size problem. From the result, we can also see that the order in magnitude among $T^d(x^{*d})$, $T^m(x^{*m})$ and $T^s(x^{*s})$ is consistent with Proposition 1.

Table 5.7: Comparison among $T^d(x^{*d})$, $T^m(x^{*m})$ and $T^s(x^{*s})$ (in minutes)

Total expected propagated delay	N_1	N_2
$T^d(x^{*d})$	10.84	335.48
$T^m(x^{*m})$	415.20	753.16
$T^s(x^{*s})$	510.06	1055.55

We now have four candidate assignments: FIFO (written as x_{FIFO}), x^{**d}, x^{**m} and x^{**m}. We compare their performance if they are used in actual practice. We know that both primary and propagated delays are random in the actual operations process. Hence we measure the performance of x^{**d} and x^{**m} under the stochastic setting, that is, using the cost matrix $c^s(x)$, and compare them with x^{**s}. Clearly, since x^{**s} is optimal for $c^s(x)$, it must outperform the others.

Further, we compare $T^s(x_{FIFO})$, $T^s(x^{*d})$, $T^s(x^{*m})$ and $T^s(x^{*s})$ on the numerical data from the two networks. The results are shown in Table 5.8. From the table, we see that $T^s(x^{*s}) \leq T^s(x_{FIFO})$, and $T^s(x^{*s}) \leq T^s(x^{*m}) \leq T^s(x^{*d})$ holds for both networks even though the difference between $T^s(x^{*m})$ and $T^s(x^{*s})$ is small, but the difference maybe big for other networks. This is consistent with the order in magnitude among $T^s(x_{FIFO})$, $T^s(x^{**d})$, $T^s(x^{**m})$ and $T^s(x^{**s})$ as well.

Table 5.8: Comparison among $T^s(x_{FIFO})$, $T^s(x^{*d})$, $T^s(x^{*m})$ and $T^s(x^{*s})$ (in minutes)

Total expected propagated delay	N_1	N_2
$T^s(x_{FIFO})$	569.84	1078.58
$T^s(x^{*d})$	569.84	1085.97
$T^s(x^{*m})$	514.94	1056.35
$T^s(x^{*s})$	510.06	1055.94

In the computation presented above, we assume that the primary delay distribution is given. For example, when we work out the aircraft assignment in August, we assume we know its distribution in August. However, in practice, before making the aircraft assignment in August, we can only use the delay data before August, and the scheduled arrival and departure times in August. In this case, there are two intuitive approaches we can use. The first approach is to use the FIFO assignment at each airport in August directly based on the scheduled arrival and departure times. Then we can derive the total expected propagated delay under the FIFO assignment by using the primary delay in August. Denote this by $T^s(x_{FIFO}, Aug)$. The second approach involves estimating the primary delay distribution for each flight by using the data in July, and using this estimated distribution to derive the optimal assignment under c^s. The we apply this assignment into the cost matrix derived from the observed primary delay in August. We denote the corresponding total expected propagated delay by $T^s(x_{Jul}, Aug)$. The comparison between $T^s(x_{FIFO}, Aug)$ and $T^s(x_{Jul}, Aug)$ is shown in Table 5.9.

Table 5.9: Comparison between $T^s(x_{FIFO}, Aug)$ and $T^s(x_{Jul}, Aug)$

Network	$T^s(x_{FIFO}, Aug)$	$T^s(x_{Jul}, Aug)$
N_1	569.84	590.00
N_2	1078.58	1073.65

From Table 5.9, we see that if we use x_{Jul} in August, it performs worse than x_{FIFO} for N_1, and marginally better than x_{FIFO} for N_2. Overall, x_{Jul} does not perform well compared to x_{FIFO}. This implies that our estimation of primary delay distribution in August using the data in July in the naive manner described here is not very effective. We need to improve the accuracy of the estimates. Hence, we explore a data-driven approach to derive an assignment so that it can perform better than both x_{FIFO} and x_{Jul}, which can serve as two benchmark policies for further comparison. We report the results of this exploration in the next section.

5.3 Data-driven Approach

In this section, we propose a data-driven approach in the deterministic, mixed and stochastic cases, respectively, to estimate the primary delay distribution, and then derive the assignments based on the distribution.

5.3.1 Data-driven Approach Under Stochastic Case

We first illustrate the data-driven approach in the stochastic case. Clearly, the central component is the estimation of primary delay distribution for each flight. Currently, most of the researches (Lan et al. 2006; Dunbar et al. 2012, 2014) do not try to estimate the primary delay distribution. By contrast, we propose a data-driven approach to estimate the primary delay distribution for each flight so that it can bring a verifiable improvement in total expected propagated delay when it is applied to the aircraft assignment problem. To illustrate this approach, we use the primary delay data in July and August in year 2007 of networks N_1 and N_2 specified in the previous section. Specifically, we first cluster eight largest primary delays among 31 primary delays in July for each flight, and then choose the optimal number of clusters based on the performance of each number of clusters. This produces the primary delay distribution to derive the assignment. Finally, we test its performance in August.

Consider the daily flight network defined in Section 9.2 with M flights in the network. There is a set of random scenarios Ω in July, where each scenario $\omega \in \Omega$ corresponds to primary delay values X^w for each flight. Let $X_i = [X_i^1, X_i^2, \cdots, X_i^{|\Omega|}]$ be a vector of primary delay values for flight i, where X_i^ω is the primary delay value for flight i under scenario $\omega \in \Omega$.

For any vector $z \in R^{|\Omega|}$, let

$$z^{[1]} \geq ... \geq z^{[|\Omega|]}$$

denote its components in decreasing order, and let

$$z^{[]} = [z^{[1]}, ..., z^{[|\Omega|]}]$$

be the z vector with components arranged in decreasing order. Thus, $X_i^{[]}$ is a permutation of X_i vector with the components arranged in decreasing order.

Let $\delta(i) = [X_i^{[1]}, X_i^{[2]}, \cdots, X_i^{[8]}]$ for $i = 1, 2, \cdots, M$. Thus, the M flights yield the M data points $(i, \delta(i)), 1 \leq i \leq M$. We analyzed these data points by using several unsupervised learning methods: the hierarchical clustering method, Gaussian mixture models, and self-organizing maps, k-means and k-medoids clustering methods. We concluded that the k-medoids clustering method works the best in reducing the total expected propagated delay. The comparison among these unsupervised learning methods is shown in Table 5.10.

Table 5.10: Comparison on the Total Expected Propagated Delay (in minutes) between Different Methods

Network	Hierarchical clustering	Gaussian mixture models	Self-organizing maps	k-means	k-medoids
N_1	580.52	595.06	590.74	605.90	536.87
N_2	1071.06	1067.23	1066.61	1071.61	1052.68

We use k-medoids clustering method to cluster $\delta = \{\delta(i), 1 \leq i \leq M\}$ into k clusters, where k is a given integer. It aims to minimize the sum of distances between each point and the center of the cluster the point lies in. The center (also called medoid) of the cluster is a member of the cluster, which is chosen as a point resulting in the smallest within-cluster distance. To be precise, for points (1-by-n vectors) y_1, y_2, ..., y_m in one cluster, the medoid of the cluster is

$$\underset{y_s, 1 \leq s \leq m}{\arg\min} \sum_{t=1}^{m} d(y_s, y_t),$$

where $d(y_s, y_t)$ is the distance between y_s and y_t. This is different from k-means clustering method since k-means clustering method chooses the sample mean of all the points in the cluster as the center.

Here, we use the Pearson correlation distance, defined as follows: .

$$d(y_s, y_t) = 1 - r = 1 - \frac{(y_s - \bar{y_s})(y_t - \bar{y_t})'}{\sqrt{(y_s - \bar{y_s})(y_s - \bar{y_s})'}\sqrt{(y_t - \bar{y_t})(y_t - \bar{y_t})'}},$$

where $\bar{y_s} = \frac{1}{n}\sum_{i=1}^{n} y_{si}$, and $\bar{y_t} = \frac{1}{n}\sum_{i=1}^{n} y_{ti}$. That is, $d(y_s, y_t)$ is equal to one minus the sample correlation r between y_s and y_t. Thus, $d(y_s, y_t)$ lies between 0 (when $r = 1$) and 2 (when $r = -1$). When $d(y_s, y_t) = 0$, it implies there is a strong positive correlation between y_s and y_t. When $d(y_s, y_t) = 2$, it implies there is a strong negative correlation between y_s and y_t. The resulting clustering has the property that the data points in the same cluster have more positive correlations. We have also tried the squared Euclidean distance, but it performs worse in reducing the total expected propagated delay than the Pearson correlation distance. The comparison is shown in Table 5.11.

Table 5.11: Comparison on the Total Expected Propagated Delay (in minutes) between the Squared Euclidean Distance and Pearson Correlation Distance under k-medoids Clustering Method

Network	Squared Euclidean distance	Pearson correlation distance
N_1	608.87	536.87
N_2	1072.35	1052.68

Thus, clustering algorithm produces a cluster function

$$cluster : \{1, 2, \cdots, M\} \to \{1, 2, \cdots, k\}.$$

Under this cluster function, flight i belongs to cluster $CL_i = cluster(i)$. We call CL_i as the clustering label of flight i.

Now for $u \in \{1, 2, \cdots, k\}$, we define

$$F(u) = \{1 \leq i \leq M : CL_i = u\}$$

as the set of flights with clustering label u. We compute $\beta^{\omega}(u)$, the ω-th sample average of the primary delays in scenario ω of flights in $F(u)$ in July. Let $\beta(u) = [\beta^1(u), \beta^2(u), \cdots, \beta^{|\Omega|}(u)]$ be a vector for a given u. Thus, the estimated primary delay value in scenario ω for flight $i \in F(u)$ is $\beta^{\omega}(cluster(i))$.

Applying the estimated primary delay for each flight under k clusters, the estimated expected propagated delay of assigning incoming flight $i \in INC(p)$ to outgoing flight $j \in OUT(p)$ under assignment x $(1 \leq p \leq P)$ is given by

$$\hat{c}_{ij}^{s}(x,k) = E(\hat{C}_{ij}^{s}(x,k)) \approx \sum_{\omega \in \Omega} \frac{1}{|\Omega|} \hat{C}_{ij}^{s,\omega}(x,k), \tag{5.29}$$

where $\hat{C}_{ij}^{s,\omega}(x,k)$ is calculated based on the estimated primary delay scenarios derived from k clusters.

Next we find the optimal number of clusters k that performs the best in July, where we assume $1 \leq k \leq 10$. That is, we first solve AP with cost matrix $c^s(x,k) = [c_{ij}^s(x^s,k)]$ by using the iterative algorithm. Let the solution derived from the iterative algorithm be $x^{*s}(k)$. Then we compute the total expected propagated delay $T^s(x^{*s}(k), Jul)$ by applying $x^{*s}(k)$ to the cost matrix derived from Equation 5.18 by using the primary delay scenarios in July. We further define the optimal number of clusters k as

$$k^{*s} = \text{argmin}\{T^s(x^{*s}(k), Jul) : 1 \leq k \leq 10\}. \tag{5.30}$$

Then we test $x^{*s}(k^{*s})$ on the primary delay scenarios in August. The corresponding total expected propagated delay when the assignment $x^{*s}(k^{*s})$ is applied to the cost matrix derived from the primary delay scenarios in August is $T^s(x^{*s}(k^{*s}), Aug)$.

5.3.2 Data-driven Approach Under Mixed Case

For the data-driven approach in mixed case, compared to the approach in stochastic case, we only consider the expected propagated delay when the delay is propagated to the next flight. That is, we change Equation 5.29 to

$$\hat{c}_{ij}^{m}(x,k) = E(\hat{C}_{ij}^{m}(x,k)) = \sum_{\omega \in \Omega} \frac{1}{|\Omega|} \hat{C}_{ij}^{m,\omega}(x,k). \tag{5.31}$$

Then we follow the same approach in the stochastic case to derive the optimal number of cluster k^{*m}, the assignment $x^{*m}(k^{*m})$, and the total expected propagated delay $T^s(x^{*m}(k^{*m}), Aug)$ in August.

5.3.3 Data-driven Approach Under Deterministic Case

For the data-driven approach in deterministic case, compared to the approach in stochastic case, we only consider the expected primary delay when the delay is propagated to the next flight. That is, we change Equation 5.29 to

$$\hat{c}_{ij}^{d}(x,k) = E(\hat{C}_{ij}^{d}(x,k)) = \sum_{\omega\in\Omega} \frac{1}{|\Omega|} \hat{C}_{ij}^{d,\omega}(x,k). \tag{5.32}$$

Then we can follow the same approach in the stochastic case to derive the optimal number of cluster k^{*d}, the assignment $x^{*d}(k^{*d})$, and the total expected propagated delay $T^s(x^{*d}(k^{*d}), Aug)$ in August. We further compare the assignments derived from data-driven approaches under these cases as well as the benchmark policies in the next section.

5.4 Computational Experiment

In this section, we first show the details on how we derive the assignments by using the data-driven approach. Then we compare the benchmark policies defined in Section 5.2 with the assignments derived from the data-driven approach using several different criteria.

5.4.1 Assignments Derived from the Data-driven Approach

In this subsection, we first show how can we derive the optimal number of clusters in the data-driven approach under deterministic, mixed and stochastic cases. We vary the number of clusters k from 1 to 10 for the data-driven approach under different cases. The result is shown in Tables 5.12 and 5.13.

Table 5.12: The Total Expected Propagated Delay (in minutes) in July with the Change of Number of Clusters k for Network N_1

k	1	2	3	4	5	6	7	8	9	10	k^*
$T^s(x^{*d}(k), Jul)$	**846.77**	846.77	846.77	846.77	846.77	846.77	846.77	846.77	846.77	846.77	1
$T^s(x^{*m}(k), Jul)$	891.58	830.23	830.61	860.48	851.03	849.71	846.71	**829.94**	847.68	842.77	8
$T^s(x^{*s}(k), Jul)$	867.45	819.68	829.58	843.65	833.77	807.74	820.55	828.58	810.48	**775.90**	10

Tables 5.12 and 5.13 show the performance of different number of clusters for the deterministic, mixed and stochastic cases for networks N1 and N2. The bold entries show the performance of the

Table 5.13: The Total Expected Propagated Delay (in minutes) in July with the Change of Number of Clusters k for Network N_2

k	1	2	3	4	5	6	7	8	9	10	k^*
$T^s(x^{*d}(k), Jul)$	1544.61	1539.10	**1538.58**	1539.06	1542.35	1546.10	1546.10	1551.00	1550.52	1550.55	3
$T^s(x^{*m}(k), Jul)$	1536.52	1620.13	1639.81	1621.39	**1496.23**	1496.23	1496.23	1496.23	1535.35	1498.35	5
$T^s(x^{*s}(k), Jul)$	1619.10	1619.10	1638.94	1621.39	1511.35	**1495.81**	1495.77	1495.81	1534.90	1495.84	6

optimal number of clusters. The corresponding cluster label for each flight in N_1 and N_2 is shown in Tables B.1 and B.2, respectively, in Appendix B. From Table B.1, we see that flights with cluster labels 4, 5 and 10 have large primary delays. For these flights, the scheduled departure/arrival times are mostly in peak hours (afternoon and night) at the originating/destination airport, and the destination airport are mostly big hubs. On the other hand, the flights with label 5 have small primary delays. Their scheduled departure/arrival times are usually in the off-peak hours (morning) at the originating/destination airport. For their destination airports, there are many non-hub airports. For other flights in N_1, they are in between. Similarly, from Table B.2, we can see that flights with labels 2 and 3 have large primary delays, and they fly during the peak hours. By contrast, the flights with labels 1 and 4 have small primary delays, and fly in the off-peak hours. The primary delays for other flights in N_2 are in between. From these two tables, we see that the clustering method tries to put the flights with similar primary delay into one cluster. The clustering result reflects the effect from originating/destination airport, and the scheduled arrival/departure time on the primary delay.

Now we apply this optimal number of clusters to determine the assignments $x^{*d}(k^{*d})$, $x^{*m}(k^{*m})$ and $x^{*s}(k^{*s})$ in August and derive the total expected propagated delay $T^s(x^{*d}(k^{*d}), Aug)$, $T^s(x^{*m}(k^{*m}), Aug)$ and $T^s(x^{*s}(k^{*s}), Aug)$ under different cases in August. These are displayed in in Table 5.14. From the table, we see that stochastic case performs the best over-all, even though the mixed case performs 1 minute better than the stochastic case for N_2.

Table 5.14: Total Expected Propagated Delay (in minutes) in August Under Different Cases

Network	N_1	N_2
$T^s(x^{*d}(k^{*d}), Aug)$	569.84	1074.84
$T^s(x^{*m}(k^{*m}), Aug)$	571.19	1051.74
$T^s(x^{*s}(k^{*s}), Aug)$	536.87	1052.68

5.4.2 Comparison

We further compare the benchmark policies, namely, x_{FIFO} and x_{Jul}, with the assignment derived from data-driven approach under the stochastic case on the total expected propagated delay, which is shown in Table 5.15. From the table, we see that $x^{*s}(k^{*s})$ performs 6.14% and 2.46% better than x_{FIFO} for N_1 and N_2, respectively. Also, $x^{*s}(k^{*s})$ performs 9.00% and 1.95% better than x_{Jul} for N_1 and N_2, respectively. From the comparison, we see that the assignment derived from data-driven approach in the stochastic case significantly outperforms the benchmark policies. Hence we recommend this approach to the airlines. This helps the airline save cost significantly by reducing the total expected propagated delay.

Table 5.15: Comparison on the Total Expected Propagated Delay (in minutes) between the Benchmark Policies and the Data-driven Approach Under the Stochastic Case

Network	N_1	N_2
$T^s(x_{FIFO}, Aug)$	569.84	1078.58
$T^s(x_{Jul}, Aug)$	590.00	1073.65
$T^s(x^{*s}(k^{*s}), Aug)$	536.87	1052.68
$\frac{T^s(x_{FIFO},Aug)-T^s(x^{*s}(k^{*s}),Aug)}{T^s(x_{FIFO},Aug)} \times 100\%$	6.14%	2.46%
$\frac{T^s(x_{Jul},Aug)-T^s(x^{*s}(k^{*s}),Aug)}{T^s(x_{Jul},Aug)} \times 100\%$	9.00%	1.95%

Further, we compare the percentage of delayed flights between the benchmark policies and the data-driven approach under the stochastic case. Based on the definition from DOT, a flight is considered to be delayed when its actual departure time is at least 15 minutes later than its scheduled departure time. We denote $pd(x, 15)$ as the percentage of flights delayed by more than 15 minutes under assignment x. If we consider 0 minute instead of 15 minutes in the delay definition, we denote $pd(x, 0)$ as the percentage of flights delayed by a positive amount. The comparison is shown in Table 5.16. From the table, we see that percentages of delayed flights for the assignments derived from the benchmark policies and the data-driven approach under the stochastic case are almost the same even though $x^{*s}(k^{*s})$ can perform kind of better than the benchmark policies. It implies that the data-driven approach under the stochastic case can decrease the delay propagation almost without influencing the percentage of delayed flights.

Table 5.16: Comparison on the Percentage of Departure Delay between the Benchmark Policies and the Data-driven Approach Under the Stochastic Case

Network	N_1		N_2	
Percentage of delay	$pd(x,0)$	$pd(x,15)$	$pd(x,0)$	$pd(x,15)$
x_{FIFO}	8.64%	6.36%	15.03%	10.97%
x_{Jul}	9.01%	6.45%	14.78%	10.92%
$x^{*s}(k^{*s})$	8.58%	6.27%	14.86%	10.92%

5.5 Conclusions

In this project, we formulate the aircraft assignment problem from a leg-based perspective by considering a balanced assignment problem between the incoming and outgoing flights at each airport in the network. Then we propose an iterative algorithm under the deterministic, mixed and stochastic cases, respectively, to solve the aircraft assignment problem. We also compare the iterative algorithm with the approach proposed by previous researches, and compare the iterative algorithm under different cases. Further, we propose a data-driven approach under different cases to estimate the primary delay distribution, which is used to derive the aircraft assignment in the future operations. Finally, we compare the assignment derived from the data-driven approach with the benchmark policies over different criteria.

We show that the iterative algorithm can bring a significant improvement in total expected propagated delay compared to the algorithms suggested by previous researchers, such as (Dunbar et al., 2014) and (Yan and Kung, 2016), by using the real-world data from one major airline. More importantly, the iterative algorithm can also reduce the computation time by orders of magnitudes compared to the previous algorithms. We further compare the iterative algorithm under different cases. We show that the optimal assignment derived from the cost matrix under the deterministic and mixed cases underestimate the total expected propagated delay significantly compared to the optimal assignment derived from the stochastic case. For the assignment derived the iterative algorithm under different cases, similar phenomenon can be observed as well by using the real-world data.

We also compare the assignments derived from the data-driven approach under different cases. We show that the data-driven approach under the stochastic case works the best. We also analyze

the clustering label for each flight using the optimal number of clusters under the stochastic case. From the analysis, we see that the clustering method tries to put the flights with similar primary delay into one cluster, which takes into account the effect from originating/destination airport, and the scheduled arrival/departure time on the primary delay. Then we compare the assignment derived from the data-driven approach under the stochastic case with the benchmark policies. We show that the assignment derived from the data-driven approach can bring a significant improvement on the total expected propagated delay compared to the benchmark policies almost without influencing the percentage of departure delays.

CHAPTER 6
Introduction

In the past three decades, the number of companies that offer service via telephone or other online tool has increased dramatically. Based on a report from CustomerServ in 2017, there are over 3.4 million call center agents employed in the United States. This phenomenon is further catalyzed by the pandemic. During the pandemic, the companies are more likely to provide the remote service through telephone or other online tool. It is estimated that the worldwide call and contact center market will reach $481 billion by 2024.

A central challenge in designing and managing a service operations in general, and a call center in particular, is to achieve the a balance between *operational efficiency* and *service quality* (Mandelbaum and Zeltyn 2009). Traditional literature on the service operations only considers admission control aspect of this problem, such as (Naor, 1969) and (Koçağa and Ward, 2010). Admission control allows system manager to accept some of the arriving customers, and reject the rest. Recent literature tries to consider the staffing aspect in addition to the admission control aspect of the problem, such as (Koole and Pot, 2011; Janssen and van Leeuwaarden, 2015) and (Sanders et al., 2017). The staffing level/service rate decision allows the system manager to choose the right number of servers/right value of service rate. However, they mostly assume that the holding cost is proportional to the queueing time (or waiting time). By contrast, in the first project, we focus on the case where a customer is satisfied if her queueing time is less than or equal to a prespecified upper limit, called the service level parameter. This reward structure is called the binary reward structure. We have not come across any admission control results using this reward structure. Then we consider a general reward structure towards the joint admission and service rate control for an unobservable queue in the second project.

We first introduce the first project in detail. We start with the introduction on the special binary reward structure. Considering the reward based on a queueing time threshold is highly relevant for the service provider to do admission control in the industry like healthcare in addition to call center,

and it is much more challenging and interesting to analyze. A most common example is a call center with multiple agents who answer calls from customers. Typically, a call center defines a customer as satisfied if her call is answered within a prespecified amount of time, say 20 seconds (Koole and Pot 2011). So the service level parameter is 20 seconds in this system. Another example is provided by a COVID-19 testing lab. The COVID test samples are collected from patients and brought to this lab where they are tested to see if the patients have COVID-19. The test sample from a patient has to be tested within 72 hours after it is collected. Otherwise, the sample deteriorates and the test results are unreliable and not very useful to the patient (Laboratory Corporation of America 2020). Thus the service level parameter in this example is 72 hours. As a third example, consider an emergency department where patients queue up for service. Consider the queue of the highest priority patients among those. If the queueing time of these patients is too long, patients would suffer serious repercussions (Shi et al. 2016). Thus we may set the service level parameter in such a system to be six hours, say.

In the first project, we focus on the important issue of management of these systems to maximize the number of satisfied customers at a minimum cost. We first discuss the cost and reward components of operating such a system. Assume that the manager earns a reward whenever a customer's queueing time is within the service level parameter. In the special case when the service level parameter is zero, a customer is satisfied if she enters service right away upon arrival, thus experiencing zero queueing time. The system manager also incurs cost for operating the system. For example, the costs in running a call center mainly arise from the salaries and the facilities. The costs in operating a COVID-19 testing labs mainly come from the testing machines and staff salaries. The costs in an emergency department are similarly related to the staffing level. We define the profit as the revenue from the satisfied customers minus the staffing costs involved in providing the services. The system manager wants to maximize the long run profit rate from this system.

The system manager has two tools at his disposal to achieve this goal: admission control and staffing level. Admission control allows him to accept some of the arriving customers, and reject the rest. Although the rejected customers are counted as unsatisfied customers, fewer admitted customers may lead to higher overall fraction of satisfied customers and hence enhance the revenue. There is no explicit additional cost to rejecting a customer, other than the loss of potential revenue. This is a common assumption in the current literature. The staffing level decision allows him to

choose the number of servers. Clearly, higher staffing level will lead to higher fraction of satisfied customers and hence higher revenue, but will also lead to higher cost. The manager needs to judiciously choose the right combination of the admission policy and the staffing level to maximize the profit. We call this the joint staffing and admission control problem. Considering the special revenue structure in the admission control combined with the discrete nature of the staffing level, it would make the joint staffing and admission control problem much more challenging and interesting to deal with.

Clearly, the joint staffing and admission policy decisions depend on the information we have about the system. In this project we consider three different levels of information: minimal information, partial information, and full information. In the minimal information case, we only know the system parameters, such as arrival rate, service rate and the service level parameter, and we do not know any other details about the system. The queue corresponding to the minimal information case is also called the unobservable queue. In the partial information case, we know the number of customers in the system in addition to the system parameters. The queue corresponding to the partial information case is also called the observable queue. Finally, in the full information case, we know the exact queueing time a customer will experience if she is admitted, in addition to the system parameters.

The minimal information case can occur in practice, but the partial information case is the most common one. For example, the call centers, the COVID testing labs or the emergency departments often know the number of customers in the system. The full information case is not that common, since it requires the knowledge of the service time of each admitted customer. We consider it mainly because it is expected to yield the best profit rate among the three levels of information and hence will provide a standard against which we can judge the relative importance of the other levels of information.

In the first project, we consider the combination of the three aspects mentioned above, namely, admission control, staffing problem and the level of information. To the best of our knowledge, these three aspects have rarely been dealt with in the literature in a joint fashion. There is, however, a large body of work in each area.

First, consider the admission control problem. There is a huge literature in this area and we will review a few relevant papers in more detail in the next chapter. This literature generally

assumes that the cost is proportional to the queueing time. Our revenue structure with nonnegative service level parameter is rare in the literature on queueing control. It is quite relevant in many queueing management situations, like the call centers, the COVID testing labs, or the emergency departments mentioned above.

Second, there is also a lot of research on the staffing problem, which we will review in the next chapter in more detail. It mainly focuses on Quality-and-Efficiency-Driven (QED) regime to derive the staffing level, where the utilization of server approaches one and simultaneously the number of servers approaches infinity. This approach usually does not incorporate the cost of staffing and generally leads to what are called the square-root staffing rules. There are few papers that incorporate a cost or reward structure to find the optimal staffing level. For example, (Koole and Pot, 2011) consider the partial information case and determine the optimal staffing level by considering the expected reward for each handled call, and the holding cost for each admitted customer as well as the cost for each server per unit time.

Lastly, there are a few papers that analyze queueing systems based on available information, and fewer still that compare the effect of different levels of information. For example, (Guo and Zipkin, 2007) consider the effect of information on customers' decisions to join or not in a single server queue, although they do not consider the staffing management or admission control. This paper and the paper by (Koole and Pot, 2011) are the two that are closest to our work presented here. We discuss them in more detail in the next section.

In the second project in this stream, we extend the previous project by considering a general reward structure under a general unobservable queuing system with a single server. It means we try to propose a joint admission and service rate policy in the second project instead of a joint staffing and admission control policy. Specifically, we first consider a service system with centralized decision maker. Each admitted customer produces a profit that depends on the system parameters, the decision variables, and the service cost that is proportional to the service rate. The decision maker has to decide what fraction of the arrivals to admit (a static policy), and at what rate to service the admitted customers (also a static policy). The decision maker wants to optimize the profit (revenue − cost). We illustrate it with two common revenue (or reward) structures that occur in many applications. The first one is the binary reward structure. The second reward structure arises if the service manager gets a fixed reward for each admitted customer, but the system also

incurs a cost proportional to the waiting time or queueing time of the customer. We call this the linear reward structure, to distinguish it from the binary reward structure. One can imagine any convex combination of these two revenue structures as another reward structure.

We shall show that if the revenue function satisfies a certain set of structural properties (for example, concavity in decision variables), the optimal joint admission and service control has a simple form as follows: if the per unit service cost is below a critical level (that depends on the system parameters), it is optimal to admit all the customers, and choose an optimal service rate that depends on the system parameters. If the per unit service cost is above this critical level, it is optimal to not admit any customers, in effect shutting the system down. This is a surprising results, since we do not make any assumptions about what the service system looks like. In fact, the concavity assumptions would imply an interior solution, that is, the admitted fraction is in the interval (0,1). The main reason behind this surprising optimality result is because the system manager is in charge of deciding both the admission policy and the service rate, that is, because the decision is centralized.

However, this immediately creates the question: what happens if the decision is not centralized? We consider this question next, under different non-centralized decision making scenarios. We assume the revenue earned by the system manager from servicing an admitted customer is the same as the reward earned by a customer if she decides to join. First, we consider the decentralized Stackelberg game where the system manager is the leader who sets the service rate, and all customers are followers who join if and only if their expected rewards are positive. This analysis is discussed in Chapter 9.3.1. The second scenario is a two-player Stackelberg game with the system manager as the leader and a single agent representing all the customers as the follower. This single agent controls the behavior of all the customers so that the aggregate per customer reward is maximized. This is discussed in detail in Chapter 9.3.2. The third scenario is a two-person Nash game between the system manager and the customer agent. Here no-one is leader and no-one is a follower. Each of the two players seek a Nash equilibrium policy that maximizes their own objective function. This is discussed in Chapter 9.3.3. We show that, in each of these three scenarios, optimal policy is identical to the centralized optimal policy. This is immensely useful, since this means that the customers naturally behave in a socially optimal way without needing any external mechanism to induce such a behavior.

In summary, the main contributions of this project are as follows: First, we analyze the joint admission and service rate problem for a general reward structure for a general queueing system. There are very few papers (see Stidham Jr and Weber 1989 and Adusumilli and Hasenbein 2010) considering the joint problem, and they do so for specific queuing systems with linear reward functions. Second, we show analytically that when the reward structure satisfies certain assumptions specified in Chapter 9.1, there exists a critical level such that when the per unit service cost is less than or equal to this critical level, the centralized optimal joint policy admits all the customers, and rejects all otherwise. This surprising result makes the optimal policy much easier to implement in reality. Third, we show that the centralized optimal joint policy remains optimal even when the decisions are not centralized. We consider three different models of non-centralized decision making: decentralized Stackelberg game, two-player Stackelberg game and two-player Nash game. In each case, the optimal policy is the same as the centralized optimal joint policy. In other words, the centralized optimal joint admission and service rate control policy also achieves the self-regulation of the unobservable queue. Lastly, we specifically study the binary and linear reward structures in an M/G/1 system and comment on the managerial insights through analytical as well as numerical results. We show numerically that the optimal policy has unexpected behavior in the binary reward structure. The details are in Chapter 9.5.

CHAPTER 7
Literature Review

There are three streams of literature relevant to our research: (1) admission control, (2) service capacity control, (3) multiple levels of information, and (4) equilibrium analysis. We discuss them below.

First consider the area of admission control. The recent book by (Stidham Jr, 2009) gives an overview on the research in this area. Most of the research on admission control focuses on the queueing systems under the partial information case, that is, it is assumed that the queue length is observable. Most papers prove that a threshold-type policy is optimal under certain conditions, that is, there exists a threshold such that an incoming customer is admitted if the number of customers in the system is below this threshold, and rejected otherwise. We first discuss these papers. There are only a few papers considering the admission control under the minimal or full information case, and we discuss them later.

We first review the literature on admission control under the partial information case. This review is by no means exhaustive. The earliest work on admission control is by (Naor, 1969), where the author considers an $M|M|1$ queueing system and determines the socially optimal admission control policy that maximizes the profit rate, that is, the reward earned from the admitted customers minus the expected waiting time cost of admitted customers in the queueing system. He also analyses the individually optimal strategy, where the customers decide whether to join or not in order to maximize their own expected reward. (Naor, 1969) finds that the customers may not behave in a socially optimal way, thus calling for regulation in the form of service tolls. (Yechiali, 1971) considers a $GI|M|1$ queueing system with a similar cost structure and shows that a threshold-type admission policy is socially optimal. (Chr, 1972) extends the previous research by considering multiple servers and non-linear waiting time cost function. (Ward and Kumar, 2008) consider the admission control of a $GI|GI|1$ queue with impatient customers in a heavy traffic regime. They incorporate costs associated with the customer rejection and reneging. They demon-

strate the asymptotic optimality of a threshold-type policy. (Koçağa and Ward, 2010) consider the admission control problem with multiple servers and incorporate costs arising from customer abandonment, server idleness and customer rejections. They show the optimality of a threshold-type policy and give an efficient iterative algorithm to minimize the long run average cost. (Borgs et al., 2014) consider a system where a single class of delay-sensitive customers seeks service from a single server, and the server prices its service and chooses an admission policy to maximize the revenue rate. They prove the optimality of a threshold-type policy, and derive the optimal threshold and optimal revenue in closed form. (D'Auria and Kanta, 2015) consider a two-node tandem network, and provide a similar threshold-type policy.

Next we review the literature on admission control under the minimal information case. (Mendelson and Whang, 1990) consider an $M|M|1$ queueing system with multiple user classes, where each class is represented by its delay cost per unit time, expected service time and an arrival rate that depends on the admission price. They derive a pricing mechanism to maximize the profit rate. The prices are not dependent on the queue length, which makes this a minimal information case. (Haviv and Oz, 2018) consider the minimal information model where the service provider admits a customer with a fixed probability so as to maximize the profit associated with the reward for each admitted customer and the expected waiting time cost. This is the socially optimal admission control. They also consider the individually optimal policies and regulation to make it compatible with the socially optimal policies.

Further, we review the admission control under the full information case. (Bekker and Borst, 2006) consider the admission control of a single server queue with a service rate that depends on the queueing time. A threshold-type policy admits an incoming customer if the workload is under a threshold. They show that this threshold-type policy maximizes the long run throughput under certain conditions, for example, an $M|G|1$ queue. (Liu and Kulkarni, 2006, 2008a,b) also do extensive analysis on admission control with the workload-dependent balking or reneging.

Our research contributes to this stream of research as follows. The papers on admission control mostly assume that the holding cost is proportional to the queueing time (or waiting time). However, we assume that we get a reward if the queueing time is less than or equal to a given threshold value in the first project. We have not come across any admission control results using this reward structure. Considering the reward based on a queueing time threshold is highly relevant for the

service provider to do admission control in the industry like healthcare. There is a large body of empirical research regarding it as the criterion to measure the performance in operations of a hospital, for example, (Pines et al., 2008; Liu et al., 2009; Singer et al., 2011; Patel et al., 2014), and (Shi et al., 2016). We also give extensive analytical analysis on the asymptotic properties on the long run revenue rate, and further analyze the optimal admission control policy and optimal revenue rate under each information case. In the second project, we consider a general reward structure, which includes the linear reward structure and binary reward structure as special cases. To the best of our knowledge, this is the first paper to consider such a general reward structure.

Next we focus on the second steam of literature, namely, the service capacity control. Most of them optimize the service capacity by changing staffing level. We begin with the work by (Halfin and Whitt, 1981), where they derive the celebrated square-root staffing formula for large systems. They also identify the precise asymptotic regime under which this result holds, now called the Halfin-Whitt regime, or the QED regime. (Borst et al., 2004) determine the asymptotically optimal staffing level in an $M|M|s$ queue by balancing the costs incurred by the server and the service quality in the QED regime. They consider a fixed hourly cost per server and fixed hourly waiting time cost for each customer, and show that the optimal staffing level follows the square-root staffing rule. Further, (Mandelbaum and Zeltyn, 2009) and (Armony et al., 2009) extend the previous research by considering the impatient customers.

Then we consider the joint staffing and admission control problem. There are not too many papers that consider the joint problem, and the few that do, fall under the partial information case, and concentrate on the QED regime. For example, (Janssen and van Leeuwaarden, 2015) consider a Markovian many-server system with retrials in the QED regime. Based on the target service level, they first determine the staffing level. They consider an admission control policy that admits an arriving customer with a probability that depends on the queue length. They do not consider profit or cost optimization. (Sanders et al., 2017) use a similar method to determine the staffing level based on the square-root staffing rule. They consider the rewards when no customers are waiting and no servers are idling, and the costs incurred when there are waiting customers or there are idle servers. They show that the revenue is maximized by a threshold-type policy that rejects an arriving customer when the queue length exceeds a certain threshold. Our model on the joint staffing and admission control problem is closest to the one considered by (Koole and Pot, 2011).

They consider the partial information case with expected reward for each handled call, holding cost per unit time for each admitted customer, and cost for each server per unit time. They show that the admission control is of a threshold type. They numerically derive the optimal staffing level and the optimal admission control threshold that maximizes the long run revenue rate. Our staffing cost model is the same as theirs, but our service reward model is different. We get a fixed reward if the queueing time of a served customer is less than or equal to a given threshold in the first project. We use this cost structure to derive the joint staffing and admission control policy. We also consider the minimal and full information cases that they do not consider. As far as we know, there are no papers considering the joint problem in the minimal or full information case.

Then we consider the joint admission and service rate control problem. (Stidham Jr and Weber, 1989) and (Adusumilli and Hasenbein, 2010) are the only two papers that analyze the joint admission and service rate control problem. They only consider an $M/M/1$ queue by considering the holding cost, service cost and so on. Specifically, (Stidham Jr and Weber, 1989) show that the optimal arrival rate and service rate are increasing in the number of customers in the system in their setting, and (Adusumilli and Hasenbein, 2010) propose an efficient algorithm to solve the joint problem even though they also show that the service rate is increasing in the number of customers in the system. Compared to their research, our second project in this stream considers a general reward structure under a general queueing system, and we give an optimal admission and service rate control policy analytically as well.

Next we review the literature on the analysis of queueing system under multiple levels of information. We are aware of only a few papers in this area. (Adan et al., 2018) consider the control problem of how arrivals should be routed to the service stations in a polling system with two nodes to minimize expected waiting costs. They give the individually and socially optimal routing policies under different levels of information. (Guo and Zipkin, 2007) analyze the customers' behavior in equilibrium in a single server queue with balking under three levels of information: minimal information, partial information and full information. The customers decide to join or balk based on their expected utilities, conditional on the information they have. They show how to compute the key performance measures, such as the server utilization and throughput, in the three information cases and obtain the closed-form solutions for some special cases. They also show that more information does not always improve the performance. Their work is close to our

first project in this stream since they consider the three levels of information. They use a general function of the queueing time of a customer, so our threshold based reward for customer service is included in their work. However, they do not consider staffing or admission control policies. We consider optimal admission control and staffing policies, and reach a different conclusion compared to that of (Guo and Zipkin, 2007). We show that the performance of the system improves with the additional information. The improvement from minimal to partial case is substantial, but the improvement from partial to full information tends to be minor under certain parameter space.

We also study the joint problem of maximizing revenue per server. This has the advantage of not needing the cost parameters. There are several papers that study the revenue per server, such as (Huselid, 1995; Guthrie, 2001; Chowdhury et al., 2014) and (Yadav et al., 2019), but they only analyzes the factors affecting the revenue per server and do not try to use it to find the optimal staffing level or admission control. Our approach provides an important quantification of revenue per server. We also show the connection between these two criteria (revenue per server vs. long run profit rate) in determining the optimal staffing level.

Finally, we review the literature on the equilibrium analysis on the admission control. There are two streams of research in this category, namely, Stackelberg equilibrium and Nash equilibrium. Stackelberg equilibrium has received extensive attention in the field of supply chain management, such as (Cachon and Zipkin, 1999; Dong and Rudi, 2004; Kouvelis and Zhao, 2012) and (Cho and Tang, 2013). To the best of our knowledge there are no papers discussing Stackelberg equilibrium policies in queueing systems. In their setting, they assume that the manufacturer is the leader who sets the wholesale price, and the retailers are the followers who determine their order quantities in response to the price. In the setting of our second project, the service provider is the leader who sets the service rate and the agent for the customers is the follower who decide the probability with which the customers join.

In the stream of Nash equilibrium, the concept is first proposed by (Nash Jr, 1950). In a Nash game among a finite set of players, each player aims to maximize its own benefit over its own set of strategies. A strategy is called a Nash equilibrium strategy if no player has an incentive deviate from it on its own. In the past several decades, there is an increasing interest in applying the Nash equilibrium to analyze the problem in electricity market and data envelopment analysis and so on, such as (Hu and Ralph, 2007) and (Liang et al., 2008). They all try to show the existence of a

pure-strategy Nash equilibrium. There are only a few papers focusing on the application of Nash equilibrium in admission control. They typically model the Nash game among the customers. The system manager who decides the service rate or the size of the waiting room is not part of the game. For example, (Naor, 1969; Yechiali, 1971) and (Haviv and Oz, 2016) focuses on the Nash equilibrium within customers when they decide whether to join the queue or not for an observable queue. Each customer decides whether to join the queue based on her own utility, thus they derive a Nash equilibrium for the system capacity above which they will not join the queue. On the other hand, (Edelson and Hilderbrand, 1975; Haviv, 2014) and (Haviv and Oz, 2018) analyze the Nash equilibrium between customers for an unobservable queue. The customers determine the Nash equilibrium for the admission probability.

As far as we know, there is no literature considering the Nash equilibrium or Stackelberg equilibrium between the system manager and customers even though there is some research analyzing the centralized decision case, such as (Stidham Jr and Weber, 1989), (Hassin and Haviv, 2003) and (Adusumilli and Hasenbein, 2010). However, it is common to see that the customers and service provider may make the decision simultaneously or sequentially in real operations. So, it is very important to consider the Nash equilibrium or Stackelberg equilibrium between the system manager and customers. As far as we know our second project is the first to consider such games. We show that, for our model, Nash equilibrium, Stackelberg equilibrium and centralized decision case are equivalent.

CHAPTER 8

Joint Staffing and Admission Control Under Different Levels of Information

8.1 Formulation and Preliminaries

In this section, we introduce the problem setting. We consider a queueing system where potential customers arrive according to a Poisson process with rate λ, and the service times are independent and exponentially distributed with parameter μ (i.e., mean $1/\mu$). There are two decisions to be made:

1. the staffing level (that is, the number of servers, s) to use to operate the system, and
2. a policy π that dictates which arrivals to be admitted to the system, and which ones to reject. The admitted customers stay in the system until they are served in a first-come-first-served (FCFS) fashion.

We assume that the staffing level is set only once initially, and cannot be changed with time or in response to system state. However, the admission decisions are based on the knowledge of the state of the system, which may change with time. Thus the staffing level is a design problem, while the admission policy is a control problem. Let Π be the set of admissible policies π. We shall define this more precisely later.

We next describe the costs and rewards that guide our choice of $s \geq 0$ and $\pi \in \Pi$. Let R_n be the non-negative (random) reward from the n-th customer. We assume that $R_n = 0$ if the n-th arrival is not admitted. Thus there is no explicit cost to reject a customer, other than the loss of potential revenue. Let c be the cost per unit time per server used to staff the system. Consider a system with staffing level s and admission policy π. Let $R(s, \pi)$ be the long run average reward per unit time for this system. That is,

$$R(s, \pi) = \lim_{t \to \infty} \frac{E_\pi(\sum_{n=1}^{N(t)} R_n)}{t}, \tag{8.1}$$

where $N(t)$ is the number of arrivals up to time t. We assume that this limit exists.

Suppose $E(R_n)$ is finite. Without loss of generality, we assume that $E(R_n) \leq 1$ for all n. The customers arrive at rate λ and each customer can produce at most a unit reward. The system can serve as most $s\mu$ customers per unit time. Note that we do not insist that $\lambda < s\mu$, since we can control the admissions. Hence we get the following bound:

$$R(s, \pi) \leq \min(\lambda, s\mu).$$

Next, let $P(s, \pi)$ be the long run net profit per unit time of operating this system, given by

$$P(s, \pi) = R(s, \pi) - cs.$$

Then our decision problem can be modeled as the following optimization problem:

$$\max_{s \geq 0,\ \pi \in \Pi} \quad P(s, \pi). \tag{8.2}$$

This is called the *joint staffing and admission control* problem. We solve it in two stages as follows. First, we fix the staffing level s, and solve the admission control problem. Define

$$R^*(s) = \sup_{\pi \in \Pi} R(s, \pi). \tag{8.3}$$

A policy $\pi^*(s)$ is called optimal if $R(s, \pi^*(s)) = R^*(s)$. We assume that such a policy exists. Clearly, we must have $R^*(s) \leq \min(\lambda, s\mu)$. The supremum of the long run profit per unit time of using the staffing level s is given by

$$P^*(s) = R^*(s) - cs.$$

Note that this could be negative. Then we solve the staffing problem

$$\max_{s \geq 0} \quad P^*(s). \tag{8.4}$$

Since $R^*(s)$ is bounded, $P^*(s)$ goes to $-\infty$ as $s \to \infty$. Hence $P^*(s)$ achieves its global maximum at some finite value of s. Denote the smallest such value of s by s^{**}. Let

$$\pi^{**} = \pi^*(s^{**}).$$

Thus the solution to the joint staffing and admission control problem is given by (s^{**}, π^{**}). The optimal profit rate and revenue rate under the optimal joint staffing and admission control is given by

$$P^{**} = P^*(s^{**}), \quad R^{**} = R^*(s^{**}).$$

Note that $R(0, \pi) = 0$ for all $\pi \in \Pi$, and hence $P(0, \pi) = 0$ for all $\pi \in \Pi$. Hence the above quantities are non-negative.

We can also consider another approach to optimal staffing problem that does not need c. Instead of maximizing the net profit rate, we maximize the revenue rate per server. We can think of this as the productivity of each server, and hence it makes sense to maximize it. See (Huselid, 1995; Guthrie, 2001; Chowdhury et al., 2014) and (Yadav et al., 2019). We solve

$$\max_{s \geq 1} \quad \frac{R^*(s)}{s}. \tag{8.5}$$

Since $R^*(s)$ is bounded, the above revenue per server goes to zero as $s \to \infty$. Hence the revenue per server achieves global maximum at some finite value of s. Denote the largest such value of s by s^*. Also let

$$c^* = \frac{R^*(s^*)}{s^*}.$$

The following theorem makes clear about the connection between these two optimization problems.

Theorem 6. Let s^* and s^{**} be as defined above. Then

1. $c < c^* \quad \Rightarrow \quad s^* \leq s^{**}$.
2. $c \geq c^* \quad \Rightarrow \quad s^{**} = 0$.

Proof. (i) If $c < c^*$, suppose $s < s^*$, then we have

$$\begin{aligned} P^*(s) &= R^*(s) - cs \\ &\leq c^* s - cs \\ &< c^* s^* - cs^* \\ &\leq R^*(s^{**}) - cs^{**} = P^*(s^{**}). \end{aligned}$$

Hence $P^*(s)$ is maximized at some $s \geq s^*$. Thus we have $s^* \leq s^{**}$.

(ii) If $c \geq c^*$, then for any s, we have

$$P^*(s) = R^*(s) - cs \leq c^* s - cs \leq 0.$$

However, we know that $P^*(0) = 0$. Hence $s^{**} = 0$. The result follows.

□

The last part of the above theorem prompts us to think of c^* as the server value: as long as the server cost $c < c^*$, we can operate the system at profit. If $c \geq c^*$, it is best to not operate the system at all.

In this project, we consider the following reward structure. Let $b \geq 0$ be a given scalar, and let W_n^q be the queueing time (not including service) of the n-th arriving customer. If the n-th arriving customer is not admitted, we assume that $W_n^q = \infty$. We define

$$R_n = \begin{cases} 1 & \text{if } W_n^q \leq b, \\ 0 & \text{if } W_n^q > b. \end{cases} \tag{8.6}$$

This implies that if the delay in starting the service of a customer is more than b, the service is worthless. Note that once the customer is admitted, we are obligated to serve her even if the service is worthless.

It makes sense to assume that the admission control policy π is influenced by the information we have about the state of the system. We consider three different levels of information,

1. (M): minimal information,

2. (P): partial information, and

3. (F): full information.

In the minimal information case, we only know the system parameters λ, μ, and b, but we do not know anything else about the state of the system. In the partial information case, we know the number of customers in the system at the time of arrival, in addition to the system parameters $(\lambda,\ \mu,\ b)$. In the full information case, we know the exact queueing time of incoming customer in addition to $(\lambda,\ \mu,\ b)$.

8.2 Minimal Information

In this section, we consider the minimal information case. Thus we assume that we know the system parameters (λ, μ, b), but not the time varying state of the system. We shall identify the set of admissible policies Π_M, compute the long run expected profit per unit time P_M, and study the joint staffing and admission control problem.

Since the system is stationary with Poisson arrivals and exponential service times, it suffices to restrict our attention to the stationary Markovian policies. Clearly, any stationary Markovian policy admits customers with a given probability in an independent fashion (Haviv and Oz 2018). Let a p-policy be an admission control policy that admits each customer with probability p in an independent fashion. Since we can not observe the system state, every stationary Markovian admission policy π_M must be a p-policy for some p. Suppose the staffing level is s, and a p-policy is followed. The queueing system is then an $M|M|s$ queue with arrival rate λp and service rate μ. Let

$$r = \frac{\lambda}{\mu}, \quad \rho = \frac{\lambda}{s\mu}. \tag{8.7}$$

The condition of stability is $p\rho < 1$. Let $\mathbb{A}$ be the set of values of p such that this system is stable. Then we see that

$$\mathbb{A} = \begin{cases} [0,1] & \text{if } \rho < 1, \\ [0, 1/\rho) & \text{if } \rho \geq 1. \end{cases} \tag{8.8}$$

We can think of $\mathbb{A}$ as the set of admissible policies Π_M under the minimal information case.

From the results about $M|M|s$ queues (See Kulkarni 2016), we see that the probability that all servers are busy is given by

$$C(s, rp) = \frac{(rp)^s}{s!}\left((1 - \frac{rp}{s})\sum_{n=0}^{s-1}\frac{(rp)^n}{n!} + \frac{(rp)^s}{s!}\right)^{-1},$$

and the steady state distribution of the queueing time W_q is given by

$$P(W_q > w) = C(s, rp)e^{-\mu(s-rp)w}, \quad w \geq 0.$$

Hence the long run expected revenue per unit time under the p-policy is given by

$$\begin{aligned} R_M(s, p) &= \lambda p P(W_q \leq b) \\ &= \lambda p(1 - P(W_q > b)) \\ &= \mu rp(1 - C(s, rp)e^{-\mu(s-rp)b}). \end{aligned} \tag{8.9}$$

Thus the long run expected profit per unit time is given by

$$P_M(s, p) = R_M(s, p) - cs,$$

and the joint staffing and admission control problem reduces to

$$\max_{s\geq 0, p\in\mathbb{A}} \quad \mu rp(1 - C(s, rp)e^{-\mu(s-rp)b}) - cs.$$

As described in the previous section, we solve this problem is in two stages in the next two subsections.

8.2.1 Admission Control

In this subsection, we fix an $s \geq 0$ and study the following admission control problem

$$\max_{p\in\mathbb{A}} \quad R_M(s, p) = \mu rp(1 - C(s, rp)e^{-\mu(s-rp)b}). \tag{8.10}$$

Let $p_M^*(s)$ be a value of p that maximizes the above revenue rate. Then, for a given s, the $p_M^*(s)$-policy is an optimal admission control policy.

Note that $R_M(s,p)$ depends on p only via $\hat{r} = rp$. So we can define

$$\hat{r}^* = \text{argmax}\{0 \le \hat{r} < s : \mu\hat{r}(1 - C(s,\hat{r})e^{-\mu(s-\hat{r})b})\}, \tag{8.11}$$

and

$$\hat{R}(s) = \mu\hat{r}^*(1 - C(s,\hat{r}^*)e^{-\mu(s-\hat{r}^*)b}). \tag{8.12}$$

Then it is easy to see that

$$p_M^*(s) = \min\{1, \frac{\hat{r}^*}{r}\}. \tag{8.13}$$

It is easy to see that $R_M(s,p)$ is an increasing function of s and b individually. It can be shown to be a concave function of p and r since $C(s,rp)$ is known to be increasing and convex in p and r, see (Lee and Cohen, 1983).

We next study the asymptotic properties of $p_M^*(s)$ and $R_M^*(s)$ as b, r, and s go to infinity in the following theorem.

Theorem 7. *In the minimal information case, we have*

(i)

$$\lim_{b\to\infty} p_M^*(s) = \begin{cases} 1 & \text{if } \rho < 1, \\ \frac{1}{\rho} & \text{if } \rho \ge 1, \end{cases} \tag{8.14}$$

$$\lim_{b\to\infty} R_M^*(s) = \begin{cases} \lambda & \text{if } \rho < 1, \\ s\mu & \text{if } \rho \ge 1, \end{cases} \tag{8.15}$$

(ii)

$$\lim_{r\to\infty} p_M^*(s) = 0, \tag{8.16}$$

$$\lim_{r\to\infty} R_M^*(s) = \hat{R}(s), \tag{8.17}$$

(iii)

$$\lim_{s\to\infty} p_M^*(s) = 1, \tag{8.18}$$

$$\lim_{s\to\infty} R_M^*(s) = \lambda. \tag{8.19}$$

Proof. (i) It is easy to see that $\lim_{b\to\infty} R_M(s,p) = \lambda p$, which is an increasing function of p. Hence p will tend to be the largest permissible value in $\mathbb{A}$, which is 1 when $\rho < 1$, and $1/\rho$ when $\rho \geq 1$. The limits of R_M follow from using these limits in Equation 8.10.

(ii) As $r \to \infty$, $\mathbb{A} \to \{0\}$, hence we must have $\lim_{r\to\infty} p_M^*(s) = 0$. From Equation 8.11, it follows that $rp_M^*(s) \to \hat{r}^*$. Hence the result follows.

(iii) Since $\lim_{s\to\infty} C(s, rp) = 0$, we have $\lim_{s\to\infty} R_M(s,p) = \lambda p$. The result follows by an argument similar to that in case 1.

□

8.2.2 Staffing Problem

Now consider the staffing problem, which involves computing the staffing level s that maximizes the net profit per unit time, assuming that for each s, we choose the optimal admission control policy $p_M^*(s)$. For a given staffing level s, the optimal revenue rate is

$$R_M^*(s) = R_M(s, p_M^*(s)).$$

Hence the optimal profit rate for a given s is given by

$$P_M^*(s) = R_M^*(s) - cs.$$

Hence the staffing problem reduces to the following optimization problem:

$$\max_{s\geq 0} \quad P_M^*(s) = \mu r p^*(s)(1 - C(s, rp^*(s))e^{-\mu(s-rp^*(s))b}) - cs. \tag{8.20}$$

Note that $R_M^*(s)$ is an increasing function of s, but may not be a concave function of s. In fact, it is numerically observed that there is an $\hat{s}$ such that $R_M^*(s)$ is convex for $s \in [0, \hat{s}]$ and concave for $s \in (\hat{s}, \infty)$. Using the argument in the previous section, we see that $P_M^*(s)$ achieves its global maximum at some finite value of s. Denote this optimal staffing level by s_M^{**} (if there are multiple

optima, choose the smallest). Then the optimal profit is given by

$$P_M^{**} = P_M^*(s_M^{**}),$$

and the admission probability under the optimal staffing is given by

$$p_M^{**} = p_M^*(s^{**}).$$

As described in the previous section, we can compute

$$s_M^* = \text{argmax}\{s \geq 1 : \frac{R_M^*(s)}{s}\}, \tag{8.21}$$

and define

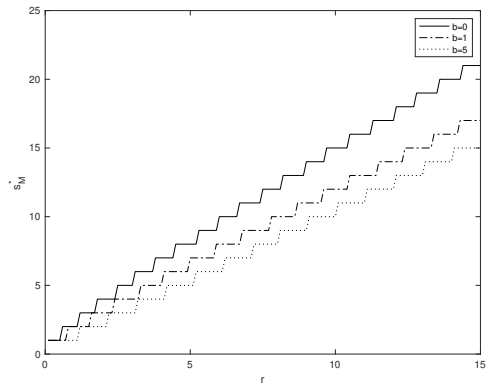
$$c_M^* = \frac{R_M^*(s_M^*)}{s_M^*}.$$

Again, choose the largest s_M^* if there are multiple optima. Theorem 6 holds with s_M^*, s_M^{**} and c_M^* in place of s^*, s^{**} and c^*.

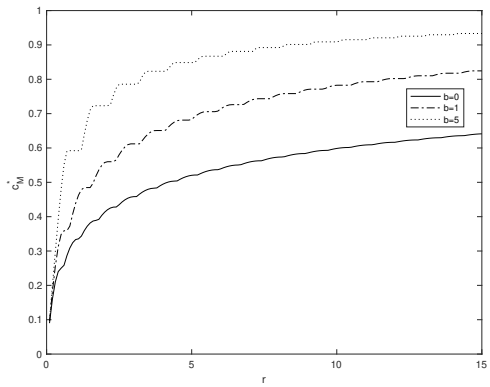
8.2.3 Numerical Results

In this subsection, we present the numerical results for the joint staffing and admission control problem. Specifically, we set $\mu = 1$ without loss of generality, and study the behavior of optimal staffing level s_M^{**}, optimal admission probability p_M^{**}, and optimal profit P_M^{**} as a function of c and r with different values of b.

We first plot s_M^* (the staffing level that maximizes worker productivity) and c_M^* (the maximum worker productivity) as a function of $r \in [0, 15]$ for $b = 0, 1, 5$ in Figures 8.1a and 8.1b, respectively. Note that c does not play any role in this computation. As expected, s_M^* increases with r and decreases with b, and c_M^* increases with r and b. c_M^* shows a non-smooth behavior, which is the result of the discrete nature of s_M^*. It also shows the right-hand limit of maximum value of c that will produce a profitable operation (using the results of Theorem 6). Thus when $r = 10$, and $b = 0$, s_M^* is 15, and c_M^* is seen to be .5986. Thus using Theorem 6, we see that if $c \geq .5986$, the optimal staffing level s_M^{**} will be zero and if $c < .5986$, s_M^{**} will be at least 15.

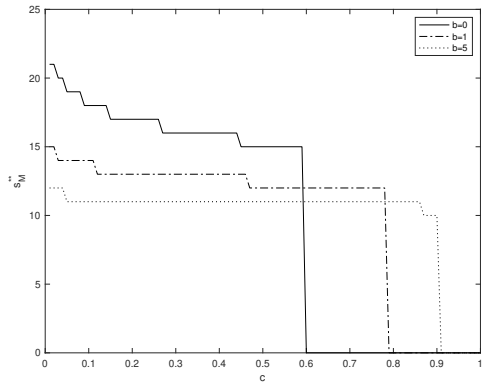


(a) s_M^* as a function of r

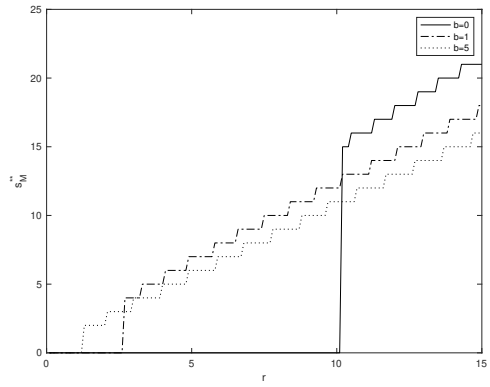


(b) c_M^* as a function of r

Figure 8.1: s_M^* and c_M^* as a function of r for different values of b under minimal information case



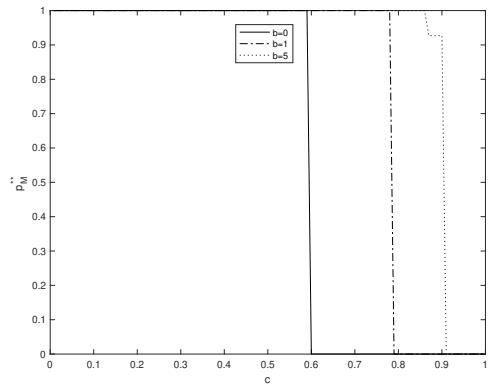
(a) s_M^{**} as a function of c



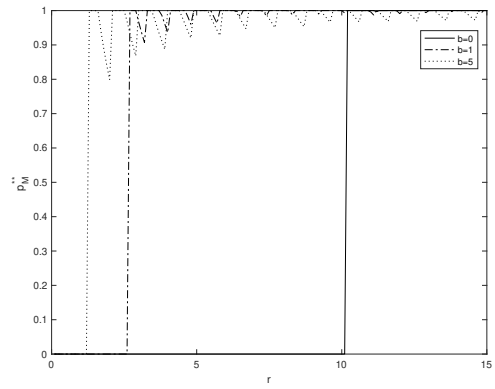
(b) s_M^{**} as a function of r

Figure 8.2: Optimal staffing s_M^{**} as a function of c and r for different values of b under minimal information case

We next show s_M^{**} as a function of $c \in [0,1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.2a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.2b. As expected, we can see that s_M^{**} is decreasing in c and b, and increasing in r. In Figure 8.2a, we see that when c crosses a threshold (depending on r and b), the server cost is too high and the optimal s_M^{**} jumps down to zero. This sudden jump is interesting. For example, consider the case of $b = 0$, the threshold is .5986. Then the optimal staffing level is 15 if c is just below .5986, and zero if it is just above .5986. This is consistent with our observation in the previous paragraph. In Figure 8.2b, we see that there is a b-dependent critical level of r below which the optimal staffing level is zero. This is because there are not enough customers to make profit since the server cost is .6 per unit time. This critical level decreases with b as expected.



(a) p_M^{**} as a function of c

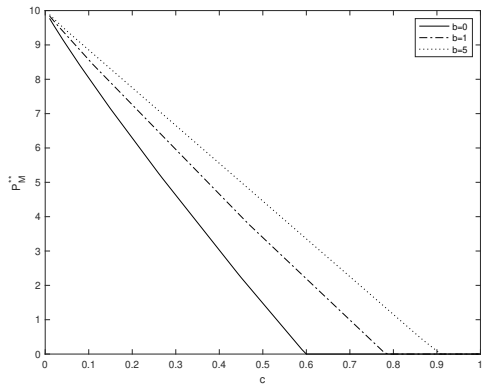


(b) p_M^{**} as a function of r

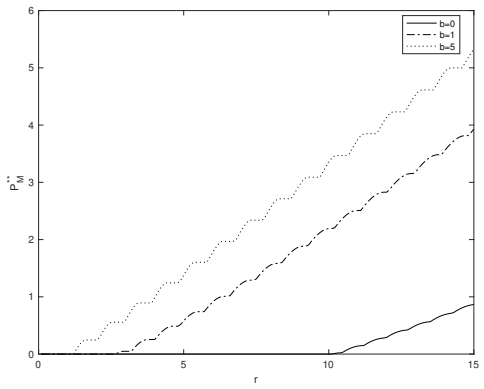
Figure 8.3: Optimal admission probability p_M^{**} as a function of c and r for different values of b under minimal information case

Further, we show p_M^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.3a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.3b. In Figure 8.3a, one expects p_M^{**} to be 1 always, since using $p_M^{**} < 1$ implies we do not make full use of incoming customers. Indeed this is true for most cases. However, in some border line cases, using $p_M^{**} = 1$ forces one to use one extra server to make more profit. Hence it may be optimal to use $p_M^{**} < 1$ if it allows one to use one less server and improve the overall profits. We can also see that $p_M^{**} = .93$ and $s_M^{**} = 10$ for $c = .87$ and $b = 5$. If one uses $p_M^{**} = 1$, it would force us to use $s_M^{**} = 11$ to maintain stability and thus adversely affect the profit. When c is small, $p_M^{**} = 1$ since we have more servers and it is optimal to admit all the customers to improve profit. When c crosses a threshold (depending on r and b), s_M^{**} and p_M^{**} both jump down to zero. In Figure 8.3b, we can see the non-monotone behavior of p_M^{**}. This occurs because s_M^{**} is discrete, and whenever it jumps up by 1, p_M^{**} shoots up to 1 to make more use of the service capacity. However, within the interval of r where s_M^{**} stays unchanged, p_M^{**} gradually decreases to avoid overcrowding and consequently degrading rewards. However, as $r \to \infty$, $p_M^{**} \to 1$ since s_M^{**} would approach infinity.

Finally, we show the optimal profit P_M^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.4a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.4b. As expected, P_M^{**} is decreasing in c, and increasing in r and b. In Figure 8.4a, we see that when c is small, we can admit all the arrivals and essentially employ enough servers to make queueing time less than or equal to b for all the customers. Hence the optimal profit approaches r as c approaches 0. This is why the three profit curves start at $r = 10$ in this figure. This is consistent with our intuition.



(a) P_M^{**} as a function of c



(b) P_M^{**} as a function of r

Figure 8.4: Optimal profit P_M^{**} as a function of c and r for different values of b under minimal information case

The profit becomes zero when $c \geq c_M^*$ and stays zero, because in this region it is optimal to not operate the system at all. The profit looks like a linear function of c in this figure, but that is only approximately true. In Figure 8.4b, we can see the non-smooth nature of P_M^{**}, which is attributable to the non-smooth nature of the admission probability p_M^{**}.

8.3 Partial Information

In this section, we consider the partial information case. Thus we assume we know the number of customers in the system at the time of arrival, in addition to parameters (λ, μ, b). We shall identify the set of admissible policies Π_P, compute the long run expected profit per unit time P_P, and study the joint staffing and admission control problem.

As in the minimal information case, we restrict our attention to stationary Markovian policies that admit or reject a customer based on the number of customers in the system at the time of arrival. One can formulate the admission control problem (for a fixed s) as a Markov Decision Process (MDP), but the reward functions involved are not convex, and hence it is not easy to derive structural properties analytically. However, if the optimal policy rejects an arriving customer at any level of the queue length, there must be a smallest value (say K) of the queue length when such a rejection occurs first. If we start the system empty (a reasonable assumption), the queue length will never exceed K under such a policy. Thus it suffices to restrict attention to a K-policy that admits the customer only when the number of customers in the system is less than K.

Now suppose the staffing level is s, and a K-policy is followed. We can assume that $K \geq s$, since it does not make sense to reject customers when there are idle servers (one can prove this rigorously, but we do not include the proof here). The queueing system is then an $M|M|s|K$ queue with arrival rate λ and service rate μ. Recall that r and ρ are defined in Equation 8.7. Note that this system is always stable for any $K < \infty$. Let p_k be the stationary probability that there are k customers in the system ($0 \leq k \leq K$). If $\rho \neq 1$, p_k is given by (see Shortle et al. 2018)

$$p_0 = \frac{1-\rho}{(1-\rho)\sum_{i=0}^{s-1} \frac{r^i}{i!} + \frac{r^s}{s!}(1-\rho^{K-s+1})},$$

and

$$p_k = \begin{cases} \frac{r^k p_0}{k!} & \text{if } 1 \leq k \leq s-1, \\ \frac{r^s p_0 \rho^{k-s}}{s!} & \text{if } s \leq k \leq K. \end{cases} \tag{8.22}$$

If $\rho = 1$, $p_k = 1/(K+1)$ for $0 \leq k \leq K$. We will find it useful to note that p_k, $0 \leq k \leq K$, are known to be decreasing and convex in K, see (Köchel, 2004).

Let $W_q(k)$ be the queueing time for an arriving customer when there are k customers in the system at the time of arrival. The FCFS service discipline implies that

$$W_q(k) = \begin{cases} 0 & \text{if } 0 \leq k \leq s-1, \\ \text{Erlang}(s\mu, k-s+1) & \text{if } s \leq k < K, \\ \infty & \text{if } k = K. \end{cases}$$

Here the equality is in distribution. Note that if the number of customers in the system is K at the time of arrival, this customer will not be admitted, and hence will produce zero reward. We indicate this by setting the queueing time of such a customer to be ∞.

Using PASTA, we see that an arrival sees the system in state k with probability p_k ($0 \leq k \leq K$). So the long run expected revenue per unit time under K-policy is given by

$$R_P(s,K) = \lambda\left(\sum_{k=0}^{s-1} p_k + \sum_{k=s}^{K-1} P(W_q(k) \leq b) p_k\right). \tag{8.23}$$

Thus the long run expected profit per unit time is given by

$$P_P(s,K) = R_P(s,K) - cs,$$

and the joint staffing and admission control problem reduces to

$$\max_{s\geq 0, K\geq s} \quad \lambda\Big(\sum_{k=0}^{s-1} p_k + \sum_{k=s}^{K-1} P(W_q(k)\leq b)p_k\Big) - cs.$$

As described in section 3, we solve this problem in two stages in the next two subsections.

8.3.1 Admission Control

In this subsection, we fix an $s \geq 0$ and study the following admission control problem

$$\max_{K\geq s} \quad R_P(s,K) = \lambda\Big(\sum_{k=0}^{s-1} p_k + \sum_{k=s}^{K-1} P(W_q(k-s+1)\leq b)p_k\Big). \tag{8.24}$$

It is worth mentioning the relevant properties of $R_P(s,K)$. As mentioned earlier, $0 \leq R_P(s,K) \leq \min(\lambda, s\mu)$. Furthermore, as $K \to \infty$, the $M/M/s/K$ queue approaches an $M/M/s$ queue. Hence we get

$$\lim_{K\to\infty} R_P(s,K) = \begin{cases} \mu r(1 - C(s,r)e^{-\mu(s-r)b}) & \text{if } \rho < 1, \\ 0 & \text{if } \rho \geq 1. \end{cases} \tag{8.25}$$

Similar to the result in the minimal information case, it can be shown that that $R_P(s,K)$ is an increasing function of s and b individually as well. Two special cases are easy to analyze: $b = 0$ and $b = \infty$. If $b = 0$, the customers who enter without delay get reward of 1, and others get a reward of zero. Hence

$$R_P(s,K) = \lambda \sum_{k=0}^{s-1} p_k, \quad (b = 0). \tag{8.26}$$

If $b = \infty$, every entering customer gets a reward of 1. Hence

$$R_P(s,K) = \lambda(1 - p_K), \quad (b = \infty). \tag{8.27}$$

It is known that $R_P(s,K)$ is a decreasing convex function of K if $b=0$, and an increasing concave function of K if $b=\infty$, for all values of ρ, see (Smith, 2003), (Köchel, 2004) and (Smith et al., 2010). In all other cases of $b\in(0,\infty)$, $R_P(s,K)$ is a unimodal function of K.

Let $K_P^*(s)$ be a value of K that maximizes the revenue rate $R_P(s,K)$. Then, for a given s, the $K_P^*(s)$-policy is an optimal admission control policy in the partial information case. Let

$$R_P^*(s)=R_P(s,K_P^*(s)).$$

Next we collect several asymptotic properties of $K_P^*(s)$ and $R_P^*(s)$ in the next theorem.

Theorem 8. *In the partial information case, we have*

(i)

$$\lim_{b\to 0}K_P^*(s)=s, \tag{8.28}$$

$$\lim_{b\to 0}R_P^*(s)=\lambda(\sum_{k=0}^{s-1}p_k), \tag{8.29}$$

(ii)

$$\lim_{b\to\infty}K_P^*(s)=\infty, \tag{8.30}$$

$$\lim_{b\to\infty}R_P^*(s)=\begin{cases}\lambda & \text{if } \rho<1,\\ s\mu & \text{if } \rho\geq 1,\end{cases} \tag{8.31}$$

(iii)

$$\lim_{r\to\infty}K_P^*(s)=s, \tag{8.32}$$

$$\lim_{r\to\infty}R_P^*(s)=s\mu, \tag{8.33}$$

(iv)

$$\lim_{s\to\infty}K_P^*(s)=\infty, \tag{8.34}$$

$$\lim_{s\to\infty}R_P^*(s)=\lambda. \tag{8.35}$$

Proof. Proof

(i) The results follow from Equation 8.26 and the fact that the revenue is a decreasing function of K.

(ii) The result about $K_P^*(s)$ follows from Equation 8.27 and the fact that the revenue is an increasing function of K. Further, when $\rho < 1$, $\lim_{b\to\infty}(1 - \frac{r^s p_0 \rho^{K-s}}{s!}) = 1$, hence $\lim_{b\to\infty} R_P^*(s) = \lambda$. When $\rho \geq 1$, $\lim_{b\to\infty}(1 - \frac{r^s p_0 \rho^{K-s}}{s!}) = 1/\rho$, hence $\lim_{b\to\infty} R_P^*(s) = s\mu$. The result now follows.

(iii) When the arrival rate goes to infinity in an $M/M/s/K$ system, if $K > s$, in the limit $s\mu$ customers enter the system, and a positive fraction of them face a positive queueing time, and hence get a reward of less than one. Hence

$$\lim_{r\to\infty} R_P(s, K) < s\mu.$$

If $K = s$, on the average $s\mu$ customers enter the system per unit time, and each of them earns a reward of 1. Hence

$$\lim_{r\to\infty} R_P(s, K) = s\mu.$$

So, when r approach infinity, $K_P^*(s) = s$.

(iv) We know $K \geq s$, so when $s \to \infty$, $K \to \infty$. Also,

$$\lim_{s\to\infty} R_P^*(s) = \lim_{K\to\infty, s\to\infty} \lambda(1 - p_K) = \lim_{K\to\infty, s\to\infty} \lambda(1 - \frac{r^s p_0 \rho^{K-s}}{s!}) = \lambda.$$

The result follows.

□

8.3.2 Staffing Problem

Now consider the staffing problem, which involves computing the staffing level s that maximizes the net profit per unit time. For a given staffing level s, the optimal revenue rate is

$$R_P^*(s) = R_P(s, K_P^*(s)).$$

Hence the optimal profit rate for a given s is given by

$$P_P^*(s) = R_P^*(s) - cs.$$

Hence the staffing problem reduces to the following optimization problem:

$$\max_{s\geq 0} \quad P_P^*(s) = \lambda\left(\sum_{k=0}^{s-1} p_k + \sum_{k=s}^{K_P^*(s)-1} P(W_q(k-s+1)\leq b)p_k\right) - cs. \tag{8.36}$$

$R_P^*(s)$ is numerically observed that there is an $\hat{s}$ such that $R_P^*(s)$ is convex for $s \in [0, \hat{s}]$ and concave for $s \in (\hat{s}, \infty)$. Using the argument in section 3, we see that $P_P^*(s)$ achieves its global maximum at some finite value of s. Denote this optimal staffing level by s_P^{**}. Then the optimal profit is given by

$$P_P^{**} = P_P^*(s_P^{**}),$$

and the optimal capacity under the optimal staffing is given by

$$K_P^{**} = K_P^*(s^{**}).$$

As described in section 3, we can compute

$$s_P^* = \text{argmax}\{s \geq 1 : \frac{R_P^*(s)}{s}\}, \tag{8.37}$$

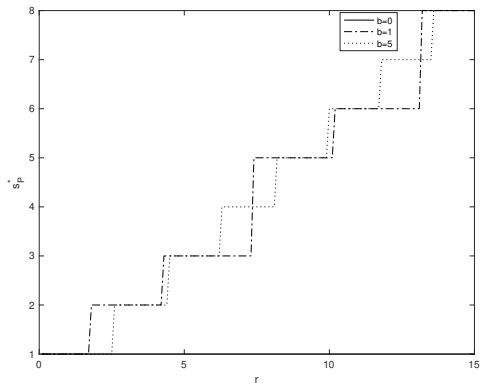
and define

$$c_P^* = \frac{R_P^*(s_P^*)}{s_P^*}.$$

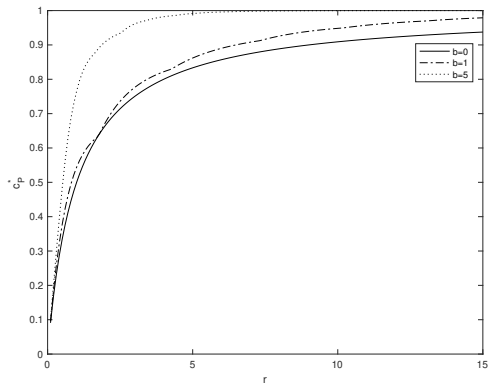
Theorem 6 holds with s_P^*, s_P^{**} and c_P^* in place of s^*, s^{**} and c^*.

8.3.3 Numerical Results

In this subsection, we present the numerical results for the joint staffing and admission control problem. Specifically, we study the behavior of optimal staffing level s_P^{**}, optimal capacity K_P^{**} and optimal profit P_P^{**} as a function of c and r with different values of b.



(a) s_P^* as a function of r



(b) c_P^* as a function of r

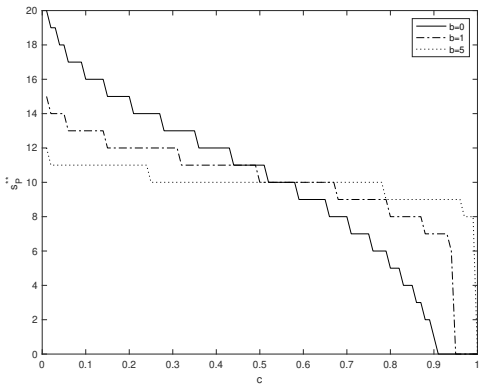
Figure 8.5: s_P^* and c_P^* as a function of r for different values of b under partial information case

We first plot s_P^* and c_P^* as a function of $r \in [0, 15]$ for $b = 0, 1, 5$ in Figures 8.5a and 8.5b, respectively. Note that c plays no role in this as well. When $b = 0$, we know from Theorem 8 that $K_P^*(s) = s$. Hence

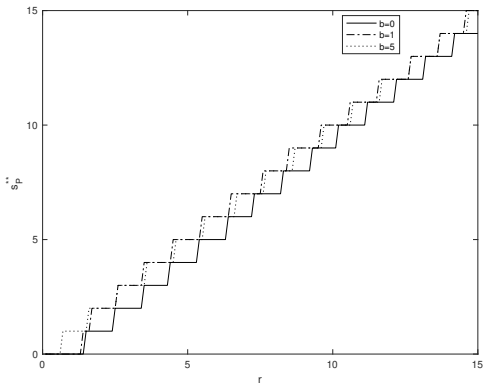
$$R_P^*(s)/s = \lambda(1 - p_K)/K.$$

This decreases in K since p_K is decreasing and convex in K (Messerli 1972). Hence $s_P^* = 1$ for all r, as seen in Figure 8.5a. When $b = 1$ and 5, we see s_P^* increases with r. But it is interesting that s_P^* is not monotone in b and its value at $b = 1$ can differ by plus or minus one from its value at $b = 5$. This is due to the discrete nature of s and K. In Figure 8.5b, we can see that c_P^* increases with r and b. Recall that c_P^{*-} (left-hand limit of c_P^*) is the maximum value of c that will produce a profitable operation (using the results of Theorem 6). Thus when $r = 10$, and $b = 0$, c_P^* is seen to be .9091. Hence if $c \geq .9091$, the optimal staffing level will be zero.

Next, we study s_P^{**}. We show s_P^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.6a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.6b. As expected, we can see that s_P^{**} is decreasing in c, and increasing in r. In Figure 8.6a, we see that when c is too high, s_P^{**} will be 0 since it is too expensive to offer the service. The behavior of s_P^{**} with b is interesting. In general, if $b_1 < b_2$, there is a parameter-dependent critical level $\hat{c}$ such that the optimal staffing level under b_1 is larger than that under b_2 for $c < \hat{c}$, and the ordering reverses if $c \geq \hat{c}$. This is due to the tradeoff involved in choosing the optimal staffing level. Intuitively, when b is larger, it is easier to get the revenue, and hence we tend to use fewer servers. If the server cost is less, we tend to use more servers under optimality. These opposing tendencies lead to the non-monotone



(a) s_P^{**} as a function of c



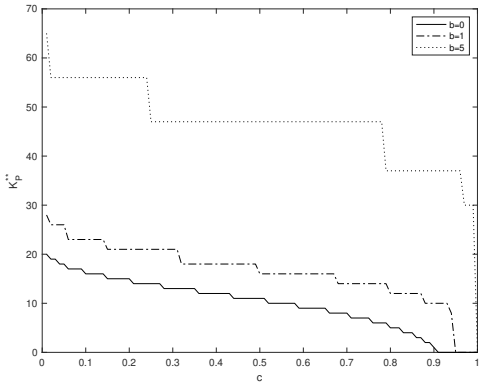
(b) s_P^{**} as a function of r

Figure 8.6: Optimal staffing s_P^{**} as a function of c and r for different values of b under partial information case

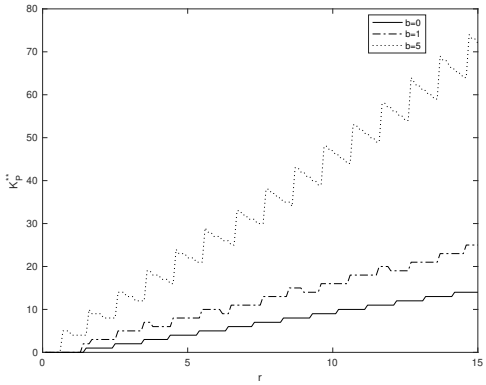
behavior of the optimal staffing level with respect to b. In Figure 8.6b , we see that the curves for $b = 1$ and 5 keep intersecting each other in many places, thus showing that s_P^{**} is not monotone in b as well. Also, note that the optimal staffing level is zero for small values of r since the arrival rate is too small to provide a profitable operation when $c = .6$.

Further, we show K_P^* as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.7a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.7b. As expected, K_P^{**} increases as b increases. In Figure 8.7a, we see that K_P^{**} gradually decreases to 0 as c increases, since optimal staffing level deceases with c and finally do not offer any service. In Figure 8.7b, we see that K_P^{**} shows non-monotone behavior with respect to r. This is the same phenomenon we observe when studying p_M^{**} as a function of r. This occurs since s_P^{**} is discrete, and whenever it jumps up by 1, K_P^{**} jumps up to exploit the added service capacity to serve more customers. However, within the interval of r where s_P^{**} stays unchanged, K_P^{**} decreases gradually since the arrival rate increases, thus reducing the service quality.

Finally, we show P_P^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.8a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.8b. As expected, P_P^{**} is decreasing in c, and increasing in r and b, in spite of the non-monotone behavior of K_P^{**}. These trends are consistent with our intuition.

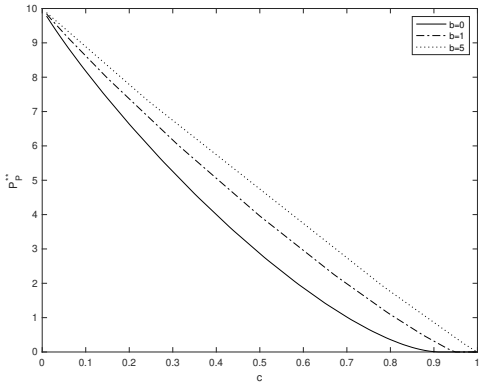


(a) K_P^{**} as a function of c

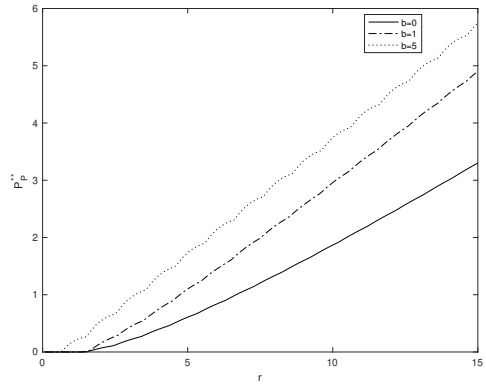


(b) K_P^{**} as a function of r

Figure 8.7: Optimal capacity K_P^{**} as a function of c and r for different values of b under partial information case



(a) P_P^{**} as a function of c



(b) P_P^{**} as a function of r

Figure 8.8: Optimal profit P_P^{**} as a function of c and r for different values of b under partial information case

8.4 Full Information

In this section, we consider the full information case. Thus we know that system parameters (λ, μ, b). In addition, we also know the exact queueing time an incoming customer faces. To make this more precise, we introduce the virtual queueing time (VQT) process in the next subsection.

8.4.1 Virtual Queueing Time Process

Let $W(t)$ be the queueing time a customer will experience if she arrives at time t and is admitted to the system. The process $\{W(t), t \geq 0\}$ is called VQT process. Let $N(t)$ be the number of customers in the system at time t. Then the definition of $W(t)$ implies that $W(t-) = 0$ if and only if $N(t-) \leq s - 1$ and $W(t+) > 0$ if and only if $N(t+) \geq s$. The system is said to be busy at time t if $N(t) \geq s$, and idle otherwise. If the system becomes busy (or stays busy) after admitting a customer, there is a jump in the $\{W(t), t \geq 0\}$ process. The sequence of jump sizes are iid $\exp(s\mu)$ random variables. When $W(t) > 0$ (thus the system is busy at time t), $W(t)$ decreases at a unit rate between upward jumps. The VQT process reaches zero (from a positive level) when exactly one server becomes idle, and stays zero until all servers become busy. Figure 8.9 shows a sample path of a system with $s = 2$ and five arrival epochs T_i $(i = 1, 2, 3, 4, 5)$. Arrivals 1, 2, 4 and 5 are admitted, while 3 is rejected. The system becomes busy at time T_1 and the busy period ends at time T' when $W(t)$ reaches 0 (simultaneously $N(t)$ reaches $s - 1 = 1$). The idle period starts at T' and ends at T_5 when $N(t)$ reaches $s = 2$ (simultaneously $W(t)$ jumps to a positive). The length of the idle period is denoted by I.

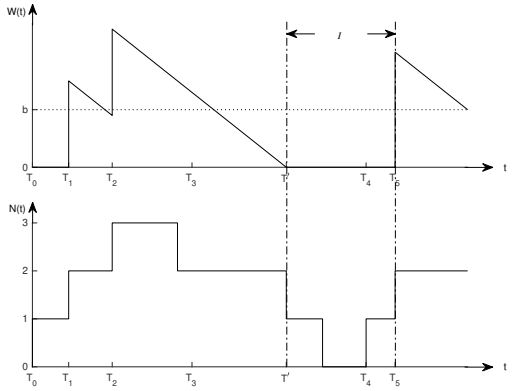


Figure 8.9: A sample path of $W(t)$ and $N(t)$

The precise definition of the idle period I is as given below:

$$I = \min\{t \geq 0 : N(t) = s | N(0) = s - 1\}.$$

We analyze it by assuming that all arrivals that see at least one idle server upon arrival are admitted (and experience zero queueing time). Let $\psi(\alpha) = E(e^{-\alpha I})$ and $\phi(\alpha)$ be equal to the total discounted reward from all the arrivals over $[0, I)$. We do not need explicit expressions for these. But we do need their expected values given below. From (Liu and Kulkarni, 2008a), we have

$$E(I) = \frac{1}{\lambda_0} = \frac{(s-1)! \sum_{k=0}^{s-1} \frac{r^k}{k!}}{r^s \mu}. \tag{8.38}$$

We can also show that the expected number of arrivals during I is given by

$$\phi(0) = \lambda E(I) - 1. \tag{8.39}$$

We study the joint staffing and admission control problem in two stages in the next two subsections.

8.4.2 Admission Control

In the full information case, we assume we know the VQT process at the time of arrival (actually its entire history up to that point). A Markovian policy decides whether to admit or reject an arrival at time t based on the VQT $W(t)$. The set of all such Markovian policies is denoted by Π_F. In this subsection, we assume that the staffing level s is fixed, and we aim to find an optimal policy $\pi_F^*(s)$ from the set of admissible policies Π_F that maximizes the long run expected revenue per unit time. We do this by deriving the optimality equation and the corresponding optimal admission control policy.

We begin by studying the discounted case first. Suppose $W(t) = w$ and a customer arrives at time t. If we admit this customer, we get a reward $r(W(t))$, defined as

$$r(w) = \begin{cases} 1 & \text{if } 0 \leq w \leq b, \\ 0 & \text{if } w > b. \end{cases} \tag{8.40}$$

Let π be an admissible policy in set Π_F. Let $v_\alpha^\pi(w)$ be the expected total discounted reward over the infinite horizon starting in state w and following policy π, using the continuous discount factor $\alpha > 0$. Suppose the n-th customer arrives at time T_n. Then we have

$$v_\alpha^\pi(w) = E_\pi\left[\sum_{n=1}^{\infty} e^{-\alpha T_n} r(W(T_n)) 1_{\{n\text{-th customer is admitted}\}} | W(0) = w)\right]. \tag{8.41}$$

Let $v_\alpha(w)$ be the optimal expected total discounted reward starting with state $W(0) = w$. Define the optimal value function as

$$v_\alpha(w) = \sup_{\pi \in \Pi_F} v_\alpha^\pi(w). \tag{8.42}$$

Now define

$$u_\alpha(w) = v_\alpha(w) - v_\alpha(b). \tag{8.43}$$

From the theory of average reward MDPs (Puterman 2014), we know that

$$g = \lim_{\alpha \to 0} \alpha v_\alpha(b) \tag{8.44}$$

exists and equals the optimal long run average revenue per unit time. Also,

$$u(w) = \lim_{\alpha \to 0} u_\alpha(w) \tag{8.45}$$

exists and represents the bias function. Then next theorem gives the optimality equation satisfied by $u(\cdot)$ and g.

Theorem 9. $u(\cdot)$ and g satisfy the following optimality equation:

$$u'(w) + g + \lambda u(w) = \lambda \max\{u(w), r(w) + \int_0^\infty u(w+t)s(t)dt\}, \quad w > 0, \tag{8.46}$$

with boundary condition

$$u'(0) = \lim_{w \to 0} u'(w) = (\frac{\lambda}{\lambda_0} - 1)(g - \lambda), \tag{8.47}$$

where

$$\lambda_0 = \frac{r^s \mu}{(s-1)! \sum_{k=0}^{s-1} \frac{r^k}{k!}}. \tag{8.48}$$

Proof. We begin by first deriving the optimality equation for $v_\alpha(w)$. Using the same argument in the partial information case, we conclude that it is optimal to admit a customer arriving at time t if $N(t) < s$. That is, if there is at least one idle server, it is optimal to admit the arriving customer. Suppose $W(0) = w > 0$. Condition on the state at an infinitesimal time h. The probability that there is no arrival over $(0, h]$ is $1 - \lambda h + o(h)$, in which case $W(h) = w - h$. The probability that there is exactly one arrival is $\lambda h + o(h)$, and if we reject this arrival, we get zero reward and the state becomes $W(h) = w - h$, and if we accept it, we get a reward of $r(w - h)$ and the state jumps to $W(h) = w - h + Y$, where $Y \sim \exp(s\mu)$ random variable. The probability of more than one arrival is $o(h)$. Combining all these cases and collecting all $o(h)$ terms, we get

$$v_\alpha(w) = (1-\lambda h)e^{-\alpha h} v_\alpha(w-h) + e^{-\alpha h}\lambda h \max\{v_\alpha(w-h), r(w-h) + \int_0^\infty v_\alpha(w-h+t)s(t)dt\} + o(h), \tag{8.49}$$

where $s(t) = s\mu e^{-s\mu t}$ is the pdf of an $\exp(s\mu)$ random variable.

Now, subtract $v_\alpha(w - h)$ from both sides, and divide by h, and let h approach zero. Then Equation 8.49 yields

$$v'_\alpha(w) + (\lambda + \alpha)v_\alpha(w) = \lambda \max\{v_\alpha(w), r(w) + \int_0^\infty v_\alpha(w + t)s(t)dt\}. \tag{8.50}$$

Next we derive the boundary condition. Suppose the idle time has started at time 0, that is $N(0) = s - 1$. Let I be the idle time. Then we admit all customers during $(0, I)$ and earn an expected discounted reward of $\phi(\alpha)$ during that time. At time I, the W process jumps to a state $Y \sim \exp(s\mu)$ at time t and earns a reward of $v_\alpha(Y)$ from then on. It is discounted by a factor $\psi(\alpha) = E(e^{-\alpha I})$. This gives us

$$v_\alpha(0) = \phi(\alpha) + \psi(\alpha) \int_0^\infty v_\alpha(t)s(t)dt. \tag{8.51}$$

Using Equation 8.43, we see that Equation 8.50 can be rewritten as

$$u'_\alpha(w) + \alpha u_\alpha(w) + \alpha v_\alpha(b) + \lambda u_\alpha(w) = \lambda \max\{u_\alpha(w), r(w) + \int_0^\infty u_\alpha(w+t)s(t)dt\}, \quad w > 0. \tag{8.52}$$

Now let $\alpha \to 0$ on Equation 8.52. Using Equation 8.44 and 8.45, we get

$$u'(w) + g + \lambda u(w) = \lambda \max\{u(w), r(w) + \int_0^\infty u(w+t)s(t)dt\}, \tag{8.53}$$

which is Equation 8.46. Letting $w \to 0$ in Equation 8.52, and using Equation 8.51, we get

$$v'_\alpha(0) + (\lambda + \alpha)v_\alpha(0) = \lambda \frac{v_\alpha(0) - \phi(\alpha)}{\psi(\alpha)}. \tag{8.54}$$

Thus when $w = 0$,

$$\begin{aligned} u'(0) = \lim_{\alpha \to 0} v'_\alpha(0) &= \lim_{\alpha \to 0} \{\lambda \frac{v_\alpha(0) - \phi(\alpha)}{\psi(\alpha)} - (\lambda + \alpha)v_\alpha(0)\} \\ &= \lim_{\alpha \to 0} \{(\frac{\lambda}{E(e^{-\alpha I})} - \lambda - \alpha)(v_\alpha(b) + u_\alpha(0)) - \frac{\phi(\alpha)\lambda}{\psi(\alpha)}\} \\ &= \lim_{\alpha \to 0} \{\alpha(\lambda E(I) - 1)(v_\alpha(b) + u_\alpha(0))\} - \lambda\phi(0) \\ &= (\lambda E(I) - 1)g - \lambda\phi(0). \end{aligned}$$

The result then follows by using the expressions for $E(I)$ and $\phi(0)$ from Equations 8.38 and 8.39. □

Now we consider a specific admission control policy called the b-policy. It admits an arrival at time t if $W(t-) \leq b$ and rejects it otherwise. Let g_b be the long run reward rate and $u_b(w)$ be the bias function of the b-policy. The next theorem gives the explicit expressions for u_b and g_b.

Theorem 10. The bias function $u_b(\cdot)$ and the average reward g_b of the $b-$policy are given by

$$g_b = \frac{-s\lambda\mu(\lambda_0 e^{\lambda b} + \lambda e^{s\mu b} - \lambda_0 e^{s\mu b} - s\mu e^{s\mu b})}{s^2\mu^2 e^{s\mu b} - \lambda\lambda_0 e^{\lambda b} - s\lambda\mu e^{s\mu b} + s\lambda_0\mu e^{s\mu b}}, \tag{8.55}$$

$$u_b(w) = \begin{cases} -g_b(w - b), & \text{if } w > b, \\ c_1 + c_2 e^{(s\mu - \lambda)w} + c_3 w, & \text{if } 0 < w \leq b, \end{cases} \tag{8.56}$$

where

$$c_1 = \frac{s\mu(\lambda - g_b)(b - e^{(s\mu-\lambda)b})}{(\lambda - s\mu)^2} - \frac{1}{s\mu - \lambda}(\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda)e^{(s\mu-\lambda)b}, \tag{8.57}$$

$$c_2 = -\frac{s\mu(\lambda - g_b)}{(\lambda - s\mu)^2} + \frac{1}{s\mu - \lambda}(\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda), \tag{8.58}$$

and

$$c_3 = \frac{s\mu(g_b - \lambda)}{\lambda - s\mu}. \tag{8.59}$$

Proof. Using the same argument as in the proof of the previous theorem, we can show that the $u_b(\cdot)$ and g_b satisfy the following optimality equation:

$$u_b'(w) + g_b + \lambda u_b(w) = \begin{cases} \lambda u_b(w), & \text{if } w > b, \\ \\ \lambda + \lambda \int_0^\infty u_b(w+t)s(t)dt, & \text{if } 0 < w \leq b, \end{cases} \tag{8.60}$$

with boundary conditions

$$u_b(b) = 0, \tag{8.61}$$

$$u_b'(b) = \lambda - g - \frac{g\lambda}{s\mu}, \tag{8.62}$$

and

$$u_b'(0) = (\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda), \tag{8.63}$$

where λ_0 is as given in Equation 8.48. Next we solve Equation 8.60.

First consider the case $w > b$. From Equation 8.60, we see that u_b satisfies the differential equation

$$u_b'(w) + g_b = 0.$$

Using the boundary condition in Equation 8.61, we see that the solution is given by

$$u_b(w) = -g_b(w - b), \quad w \geq b. \tag{8.64}$$

This is the first case in Equation 8.56.

Next consider the region $0 < w \leq b$. From Equation 8.60, we see that in this region u_b satisfies the differential equation

$$u_b'(w) + \lambda u_b(w) - \lambda \int_0^\infty u_b(w+t)s(t)dt = \lambda - g_b.$$

Substituting from Equation 8.64, this reduces to

$$u_b'(w) + \lambda u_b(w) - s\lambda\mu \int_0^{b-w} u_b(w+t)e^{-s\mu t}dt = \lambda - g_b - \frac{g_b\lambda e^{-s\mu(b-w)}}{s\mu}.$$

Now use the transformation

$$u_b(w) = e^{s\mu w}f(w) \tag{8.65}$$

in the above equation and simplify it to get

$$f'(w) + (\lambda + s\mu)f(w) - s\lambda\mu \int_w^b f(z)dz = (\lambda - g_b)e^{-s\mu w} - \frac{g_b\lambda e^{-s\mu b}}{s\mu}.$$

Taking the derivative with respect to w on both sides of the above equation and simplifying, we get

$$f''(w) + (\lambda + s\mu)f'(w) + s\lambda\mu f(w) = s\mu(g_b - \lambda)e^{-s\mu w}.$$

This is a non-homogeneous second order linear differential equation that can be solved by standard methods to obtain

$$f(w) = c_1 e^{-s\mu w} + c_2 e^{-\lambda w} + c_3 w e^{-s\mu w},$$

where c_3 is as given in Equation 8.59. The last term arises from the non-homogeneous part. Substituting the above in Equation 8.65 leads to

$$u_b(w) = c_1 + c_2 e^{(s\mu - \lambda)w} + c_3 w, \quad 0 < w \leq b,$$

which is the second part of Equation 8.56. Finally, the boundary conditions in Equations 8.61, 8.62 and 8.63, provide the three equations to determine the three unknowns c_1, c_2 and g_b. The final expressions for these three unknowns are given in Equations 8.57, 8.58, and 8.55. □

Clearly the expressions in the statement of the above theorem need to be modified if $\lambda = s\mu$. We give the result in the following corollary.

Corollary 1. In the special case when $\lambda = s\mu$, the results of Theorem 10 reduce to

$$g_b = \frac{\lambda^2(\lambda_0 b + 1)}{\lambda + \lambda_0 + \lambda\lambda_0 b}, \tag{8.66}$$

$$u_b(w) = \begin{cases} -g_b(w-b), & \text{if } w > b, \\ c_1 + c_2 w + c_3 w^2, & \text{if } 0 < w \leq b, \end{cases} \tag{8.67}$$

where

$$c_1 = -(\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda)b - \frac{\lambda(g_b - \lambda)b^2}{2}, \tag{8.68}$$

$$c_2 = (\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda), \tag{8.69}$$

and

$$c_3 = \frac{\lambda(g_b - \lambda)}{2}. \tag{8.70}$$

Using Theorem 10, we get the main result in the next Theorem.

Theorem 11. *The b-policy maximizes the long run average reward per unit time.*

Proof. We shall first show that

$$u_b(w) > \int_0^\infty u_b(w+t)s(t)dt, \text{ if } w > b, \tag{8.71}$$

and

$$u_b(w) \leq 1 + \int_0^\infty u_b(w+t)s(t)dt, \text{ if } 0 < w \leq b. \tag{8.72}$$

First consider the case $w > b$. Using the first part of Equation 8.56, we have

$$\int_0^\infty u_b(w+t)s(t)dt = -g_b(w-b) - \frac{g_b}{s\mu} = u_b(w) - \frac{g_b}{s\mu} < u_b(w).$$

This proves Equation 8.71.

Now consider the case $0 < w \leq b$. Using the second part of Equation 8.56 and 8.60, and Equation 8.55, we get

$$\begin{aligned} 1 + \int_0^\infty u_b(w+t)s(t)dt - u_b(w) &= u_b'(w) + g_b \\ &= e^{s\mu w}(s\mu f(w) + f'(w)) + g_b \\ &= \frac{s\mu(\lambda - g_b)(1 - e^{(s\mu-\lambda)w})}{s\mu - \lambda} + (\frac{\lambda}{\lambda_0} - 1)(g_b - \lambda)e^{w(s\mu-\lambda)} + g_b \\ &= \frac{-s\lambda\mu(\lambda - s\mu)}{s\mu(s\mu + \lambda_0) - \lambda(\lambda_0 e^{(\lambda - s\mu)w} + s\mu)} \geq 0. \end{aligned}$$

Hence Equation 8.72 holds.

Then we see that Equation 8.60 satisfied by u_b and g_b reduces to the optimality equations in Theorem 9. Hence g_b is the optimal long run reward rate and b-policy is optimal. □

With these results about the optimal admission control policies, we are ready to study $R_F^*(s)$, the optimal long run reward rate if the number of servers is fixed at s. Then we have proved that it is achieved by following the b-policy, and is given by g_b of Equation 8.55 if $\lambda \neq s\mu$ and g_b of Equation 8.66 if $\lambda = s\mu$. It is seen that $R_F^*(s) = g_b$ is a concave function of s for a given set of parameters b, λ, and μ, although it is difficult to prove this algebraically. For the case $s = 1$, we get

$$R_F^*(1) = \begin{cases} \frac{\rho(\lambda e^{\lambda b} - \mu e^{\mu b})}{\rho^2 e^{\lambda b} - e^{\mu b}}, & \lambda \neq \mu, \\ \\ \frac{\lambda(\lambda b + 1)}{2 + \lambda b}, & \lambda = \mu. \end{cases} \tag{8.73}$$

This result can also be derived by using the limiting distribution of the VQT process derived in (Liu and Kulkarni, 2006). Based on the expression for $R_F^*(s)$, we present its asymptotic properties in the following Theorem. The results are intuitive and the proof is purely algebraic and hence is omitted.

Theorem 12. *In the full information case, we have*

(i)

$$\lim_{b \to 0} R_F^*(s) = \lambda(\sum_{k=0}^{s-1} p_k), \tag{8.74}$$

where p_k is the stationary probability of k customers in an $M|M|s|s$ queue,

$$\lim_{b\to\infty} R_F^*(s) = \begin{cases} \lambda & \text{if } \rho < 1, \\ s\mu & \text{if } \rho \geq 1, \end{cases} \tag{8.75}$$

(ii)

$$\lim_{r\to\infty} R_F^*(s) = s\mu, \tag{8.76}$$

(iii)

$$\lim_{s\to\infty} R_F^*(s) = \lambda. \tag{8.77}$$

We can see that the asymptotic properties of $R_F^*(s)$ are the same as the asymptotic properties of $R_M^*(s)$ and $R_P^*(s)$.

8.4.3 Staffing Problem

Now consider the staffing problem, which involves computing the staffing level s that maximizes the net profit per unit time. For a given staffing level s, the optimal revenue rate is computed as $R_F^*(s) = g_b$ in the previous subsection. Hence the optimal profit rate for a given s is given by

$$P_F^*(s) = R_F^*(s) - cs.$$

Thus the staffing problem reduces to the following optimization problem:

$$\max_{s\geq 0} \quad P_F^*(s) = R_F^*(s) - cs. \tag{8.78}$$

Since $R_F^*(s)$ is a concave function of s, so is $P_F^*(s)$. Hence $P_F^*(s)$ achieves its global maximum at some finite value of s. Denote this optimal staffing level by s_F^{**}. Then the optimal profit is given by

$$P_F^{**} = P_F^*(s_F^{**}).$$

As described in section 3, we can compute

$$s_F^* = \text{argmax}\left\{ s \geq 1 : \frac{R_F^*(s)}{s} \right\}. \tag{8.79}$$

Since $R_F^*(s)$ is a concave function of s, it follows that the optimal revenue per server decreases with s, and hence $s_F^* = 1$ always! This is indeed very counterintuitive and very different from the minimal and partial information cases. Since $s_F^* = 1$, we get

$$c_F^* = \frac{R_F^*(s_F^*)}{s_F^*} = R_F^*(1),$$

where $R_F^*(1)$ is explicitly given in Equation 8.73. Theorem 6 holds with s_F^*, s_F^{**} and c_F^* in place of s^*, s^{**} and c^*.

8.4.4 Numerical Results

In this subsection, we present the numerical results for the joint staffing and admission control problem. Specifically, we study the behavior of optimal staffing level s_F^{**} and optimal profit P_F^{**} as a function of c and r with different values of b.

We skip the study of s_F^* since it is always 1. In figure 8.10, we plot c_F^* as a function of $r \in [0, 15]$ for $b = 0, 1, 5$. We can see that it increases with r and b. It also shows the right-hand limit of maximum value of c that will produce a profitable operation (using the results of Theorem 6). Thus when $r = 10$, and $b = 0$, c_F^* is seen to be 0.9091, which is the same as the corresponding value in the partial information case since they are both equivalent to an $M|M|s|s$ queue when $b = 0$. Thus if $c \geq 0.9091$, the optimal staffing level will be zero.

We next show s_F^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.11a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$, and $c = .6$ in Figure 8.11b. As expected, we can see that s_F^{**} is decreasing in c, and increasing in r. In Figure 8.11a, we see that when c is too high ($c \geq 0.9091$ when $b = 0$), s_F^{**} will be 0 since it is too expensive to offer the service. Also, for the behavior of s_F^{**} with respect to b, it is similar to that in the partial information case. In Figure 8.11b, we note that the optimal staffing level is zero for small values of r since the offered traffic is not large enough to run a profitable system at this server cost.

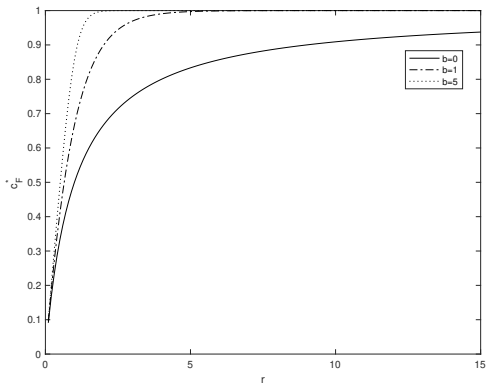
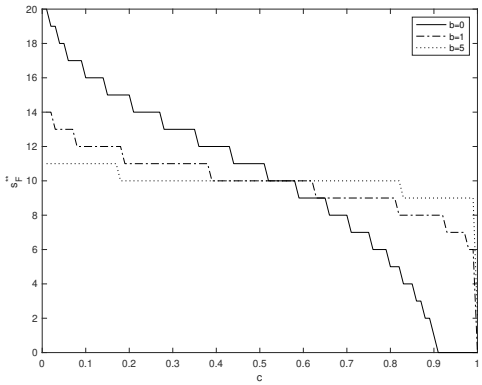
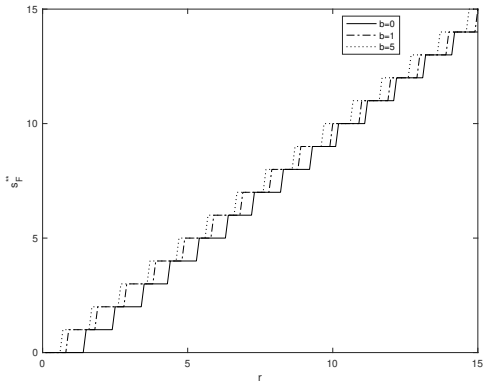


Figure 8.10: c_F^* as a function of r for different values of b under full information case



(a) s_F^{**} as a function of c



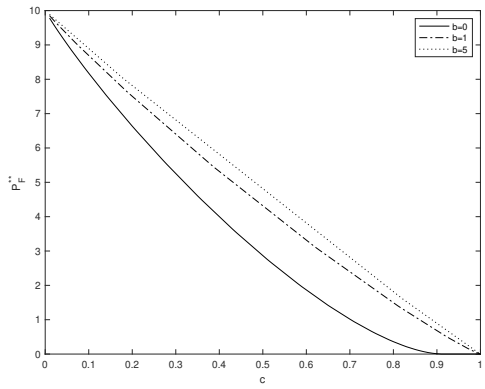
(b) s_F^{**} as a function of r

Figure 8.11: Optimal staffing s_F^{**} as a function of c and r for different values of b under full information case

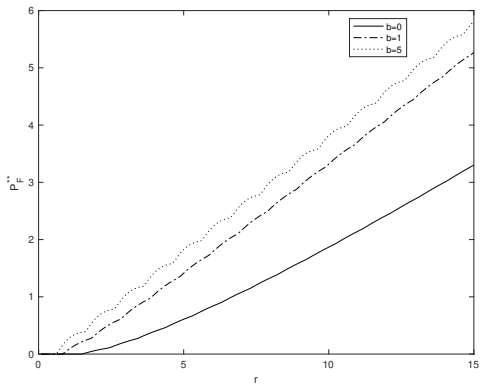
We finally show P_F^{**} as a function of $c \in [0, 1]$ for $b = 0, 1, 5$, and $r = 10$ in Figure 8.12a, and as a function of $r \in [0, 15]$ for $b = 0, 1, 5$ for $c = .6$ in Figure 8.12b. As expected, P_F^{**} is decreasing in c, and increasing in r and b.

8.5 Value of Information

In this section, we study how the available information (minimal, partial or full) affects the performance of the system under optimal operations. We know previously that when $b \to 0$, the partial and full information cases have the same joint staffing and admission control policy, which is different from that in the minimal information case. And when $b \to \infty$, all the information cases have the same joint staffing and admission control policy. What less clear is the case when b is in between. So, we choose $b = 1$ in the following analysis.



(a) P_F^{**} as a function of c



(b) P_F^{**} as a function of r

Figure 8.12: Optimal profit P_F^{**} as a function of c and r for different values of b under full information case

To begin with, we study the effect of information on server value c^*. We plot the ratios c_M^*/c_F^* and c_P^*/c_F^* as a function of $r \in [0, 15]$ in Figure 8.13, for $b = 1$. We see that

$$\frac{c_M^*}{c_F^*} \leq \frac{c_P^*}{c_F^*} \leq 1,$$

implying that the server value increases with information. In other words, we can afford to pay the workers more and still make profit if we have more information. It is also important to realize that this effect is non-monotone in r: it decreases from 1 initially and then increases to 1. Thus the server value in the partial (minimal) information case can be as low as 74% (59%) of that in the full information case. However, when r is large, the ratios approach 1, implying that the effect of extra information diminishes. This provides an important insight about the value of information.

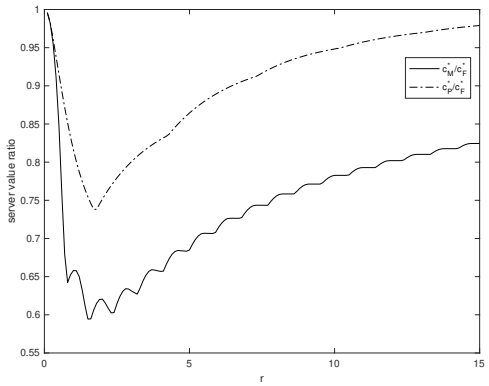
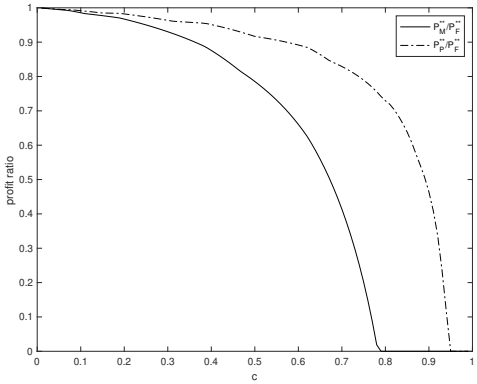


Figure 8.13: The server value ratio as a function of r

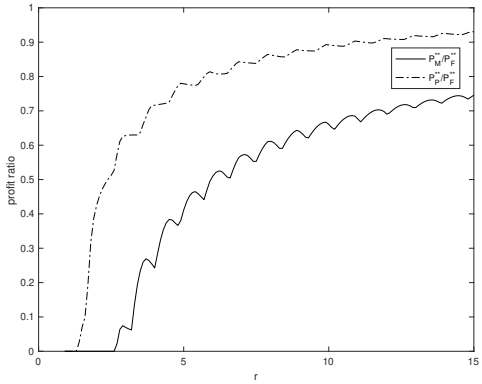
We next explore the effect of available information on the optimal profit P^{**}. We plot the ratios P_M^{**}/P_F^{**} and P_P^{**}/P_F^{**} as a function of $c \in [0,1]$ in Figure 8.14a for $r = 10$ and $b = 1$, and as a function $r \in [0,15]$ in Figure 8.14b for $c = .6$ and $b = 1$. We see that

$$\frac{P_M^*}{P_F^*} \leq \frac{P_P^*}{P_F^*} \leq 1,$$

implying that extra information leads to better profits. This is to be expected. We interpret the ratios as profit efficiency: it tells us what fraction of the profit under full information is captured by minimal and partial information levels. Figure 8.14a shows that the profit efficiency decreases with c under both information levels. The efficiency hits zero when c increases beyond the server value in each case. Another implication is that we should be willing to pay more for information if our servers are more expensive. Figure 8.14b shows that the profit efficiency tends to increase as r increases, although the effect is not strictly monotone. This is the consequence of discrete nature of the optimal staffing level and admission control. This figure also implies that the usefulness of the extra information diminishes as the traffic load increases. We also see that the optimal profit under the partial information case is not too much worse than the full information case when c is small and r is large. In this case, it may not be able to help the service provider too much by getting more information when c is small and r is large given he has partial information already.



(a) The profit ratio as a function of c



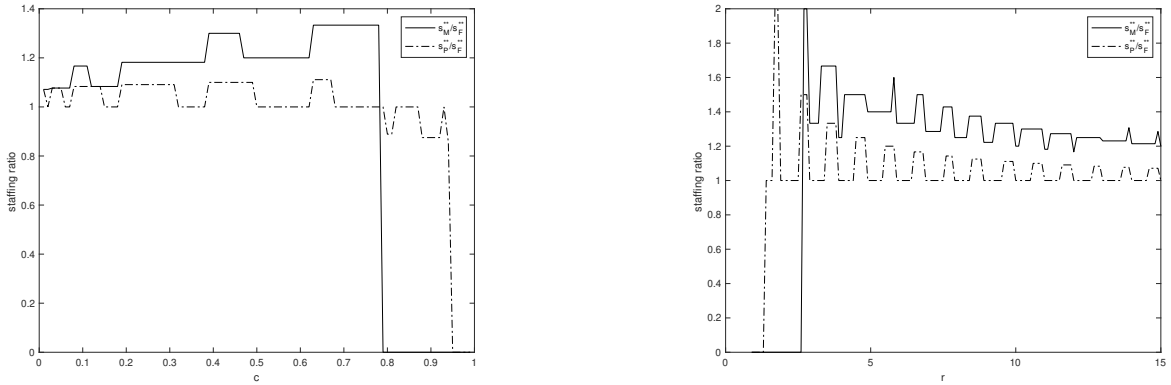
(b) The profit ratio as a function of r

Figure 8.14: The profit ratios as a function of c and r

Next we study the effect of information on the optimal staffing level s^{**}. This is not as easy to guess intuitively. We plot the ratios s_M^{**}/s_F^{**} and s_P^{**}/s_F^{**} as a function of $c \in [0,1]$ in Figure 8.15a,

for $r = 10$ and $b = 1$, and as a function of $r \in [0, 15]$ in Figure 8.15b for $c = .6$ and $b = 1$. We have seen previously that s^{**} decreases in c and increases in r in all the information cases. One might expect that the optimal staffing level will be smaller with more information. We can think of the ratio as staffing efficiency. A staffing ratio of 1.2 would mean that full information is 20% more efficient. Clearly the ratio cannot be more than one for all parameter values, since we know that the optimal staffing level drops to zero if $c \geq c^*$, and we have seen that c^* increases with information.

When $c < c^*_M$, we see that the ratio is always more than 1 for the minimal information case. That is, we use more servers under the minimal information than under the full information. The story of the partial information is less clear. In that case, we see that when $c < c^*_P$, the ratio hovers around one, and even dips below one as c approaches c^*_P. Similar effect is observed when we vary r in Figure 8.15b. In general, when the staffing levels are not zero, the optimal staffing level under the partial information case is comparable to that under the full information case, but the staffing level under minimal information is always more than that under the partial or full information case. The discontinuous nature of the graph is because the optimal staffing levels are discrete.

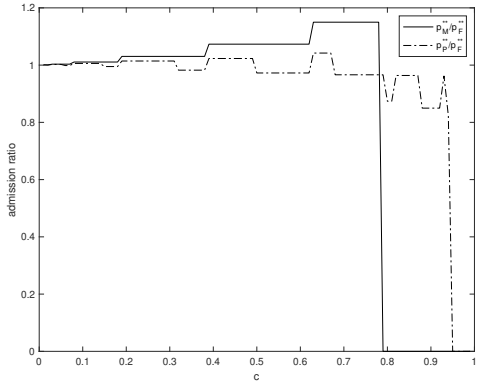


(a) The staffing ratio as a function of c **(b)** The staffing ratio as a function of r

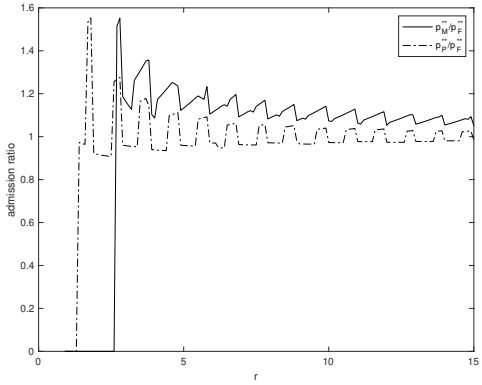
Figure 8.15: The staffing ratios as a function of c and r

Staffing level is of course intimately influenced by the admission control policy. Since the policies under different information levels have different structures (p-policy under M, K-policy under P, and b-policy under F), we need a common criterion for comparison. We choose to compare the effect of information on the fraction of the arrivals that are admitted under optimal admission policy, which we denote as p^{**}. Specifically, $p^{**} = p^{**}_M$ in the minimal information case, p^{**} equals the

stationary probability that an incoming customer enters an $M|M|s_P^{**}|K_P^{**}$ queueing system in the partial information case, and $p^{**} = R_F^{**}/\lambda$ in the full information case. We plot the ratios p_M^{**}/p_F^{**} and p_P^{**}/p_F^{**} as a function of $c \in [0,1]$ in Figure 8.16a, for $r = 10$ and $b = 1$, and as a function of $r \in [0,15]$ in Figure 8.16b for $c = .6$ and $b = 1$. From the figure, we can see that the minimal information case always allows a significantly larger fraction of customers to enter service compared to the other two information cases, leading to significantly smaller profits. It is interesting to note that the ratio in the partial information case does not always exceed one, and is less than one as c approaches c_P^*. It hovers around 1 as r increases. We have seen similar behavior for the staffing ratios. Since full information is typically much harder to implement in practice (since it involves knowing the actual service time of every admitted customer in the system), it is useful to know that the performance of the optimal staffing and admission control policies in the partial information case is not that far from the full information case when c is small and r is large. Thus using partial information can often be the most practical strategy to use in real world in that parameter space.



(a) The admission ratio as a function of c



(b) The admission ratio as a function of r

Figure 8.16: The admission ratios as a function of c and r

8.6 Conclusions

In this project, we consider a joint staffing and admission control problem under the minimal, partial and full information. Under each information case, we do this in two stages. In the first stage we fix the staffing level and determine the optimal admission control policy to maximize the revenue rate. We analyze various asymptotic properties of the optimal revenue rate under such a policy. In the second stage we determine the optimal staffing level to maximize the profit rate assuming

that we follow a staffing-level-dependent admission control policy. We also study the maximum revenue per server under each information level. Finally, we consider the effect of information on the server value, optimal profits, optimal staffing levels and optimal admission fractions under the three information cases.

In terms of the admission control under each information case, we show that the revenue is concave over the probability p under the minimal information case, and concave over the capacity K under the partial information case, and the optimal revenue under full information case is achieved under b-policy. We also observe a surprising phenomenon that the optimal admission probability under minimal information case, and optimal capacity under the partial information case shows a non-monotone behavior over the traffic level. This is because the optimal staffing level is discrete, and whenever it jumps up by 1, the optimal probability under the minimal information case, and optimal capacity under the full information case increase suddenly to make full use of the service capacity. However, when the optimal staffing level keeps the same, the optimal probability and optimal capacity would gradually decrease to avoid overcrowding with the increase of the traffic level. By contrast, the b-policy is always optimal under the full information case no matter how the traffic level changes since we have full information over the queueing time, thus do not have to worry about the use of service capacity, and overcrowding. It implies that the trends of the admission control policy under minimal and partial information cases are similar, which are very different from that of full information case.

In terms of the staffing level under each information case, we show that the optimal staffing level will become 0 when the server cost is larger than or equal to the server value. It implies if the server cost is too high, it would be too expensive to offer the service. We observe that the optimal staffing level is decreasing in the server cost and increasing in the traffic level under each information case. We also notice that the optimal staffing under minimal information case is always larger than that in the partial or full information case, but the optimal staffing level under the partial information case is comparable to that under the full information case.

In terms of the profit, the minimal information level significantly under-performs the partial and full information levels. However, the comparison between partial and full information cases is more nuanced. The partial information level only mildly under-performs the full information case when the server cost is small and the traffic level is high, but when the server cost is large or the

traffic level is low, the full information case out-performs the partial information case. It implies our model can not only help the service provider decide the optimal admission control and staffing policy based on the information he has about the system, but also help the service provider realize the potential improvement in profit if he can get additional information. For example, based on the service level parameter, server cost and traffic level, the service provider can decide whether it is worthwhile to get additional information in order to get more profit.

CHAPTER 9

Joint Admission and Service Rate Control of an Unobservable Queue

9.1 The Model

We consider a queueing system where customers arrive at an average rate λ. The system operator has two controls: the admission probability $p \in [0,1]$, and the average service rate μ. If the admission probability is p, the customers enter the system at rate λp. The system is stable if $\mu > \lambda p$. We find it more convenient to use the excess service capacity $\theta = \mu - \lambda p \geq 0$ as a control in place of μ. We call this the (p, θ) policy. Henceforth we assume

$$(p, \theta) \in S = [0,1] \times [0, \infty).$$

We treat the case $p = 0$ as a special case, since in this case no one is entering the system, so it makes sense to set $\mu = \theta = 0$ and not operate the system at all.

Suppose each admitted customer generates an expected reward $m(p, \theta)$. Then

$$f(p, \theta) = \lambda p m(p, \theta)$$

is the average reward per unit time from the admitted customers under the policy (p, θ).

Suppose the cost of choosing a service rate μ is $c\mu$ per unit time, where c is the cost per unit time of a server that serves at rate 1, called per unit server cost. Then the expected profit per unit time is given by

$$g(p, \theta, c) = f(p, \theta) - c(\lambda p + \theta).$$

We use the following notation for the partial derivatives, assuming they exist:

$$f_p(p, \theta) = \frac{\partial f(p, \theta)}{\partial p}, \quad f_\theta(p, \theta) = \frac{\partial f(p, \theta)}{\partial \theta},$$

$$f_{p,p}(p,\theta)=\frac{\partial^2 f(p,\theta)}{\partial p^2},\ \ f_{p,\theta}(p,\theta)=\frac{\partial^2 f(p,\theta)}{\partial p\partial\theta},\ \ f_{\theta,\theta}(p,\theta)=\frac{\partial^2 f(p,\theta)}{\partial\theta^2}.$$

We make the following assumptions.

Assumption 1. *(i):* $f_\theta(p,\theta)\geq 0$, *(ii):* $f_{p,\theta}(p,\theta)\geq 0$, *(iii):* $f_{p,p}(p,\theta)\leq 0$, *and (iv):* $f_{\theta,\theta}(p,\theta)\leq 0$, $p>0,\theta>0$.

Assumption 1 implies that f is concave in p, and increasing concave in θ, but not necessarily jointly concave in p and θ. The function need not be continuous at $(p,\theta)=(0,0)$, and we simply define it to be 0.

Assumption 2 is about the behavior of f at the boundaries. This is needed in Lemma 4 in Section 9.2.

Assumption 2. *(i):* $\lim_{\theta\to 0^+} f(p,\theta)\leq 0$, *and (ii):* $\lim_{\theta\to\infty} f(p,\theta)=l\in(0,\infty)$.

Finally, we need a technical assumption below.

Assumption 3. $pm_p(p,\theta)+\theta m_\theta(p,\theta)\geq 0$.

This is needed in Lemmas 5 and 6 in Section 9.2. Even though this assumption can be expressed in terms of $f(p,\theta)$ instead of $m(p,\theta)$, it is much clearer to use $m(p,\theta)$ instead.

9.2 Socially Optimal Policy

In this section we derive the socially optimal policy that maximizes the profit rate. That is, we assume there is a central decision maker, namely the system manager, who maximizes the expected profit per unit time. We assume that the system manager knows the system parameters λ and c, but does not know any other details about the state of the queuing system. He wants to choose a policy (p,θ) that maximizes the expected profit rate $g(p,\theta,c)$. Thus, he solves the following optimization problem

$$\max_{p\in[0,1],\theta\geq 0}\quad g(p,\theta,c). \tag{9.1}$$

Let the optimal solution be p^* and θ^*. We call (p^*,θ^*) the socially optimal policy (SOP). Note that $g(p^*,\theta^*,c)\geq 0$, since we can always decide not to operate the system, that is, choose $p=0$ and $\theta=0$ (equivalently, $\mu=0$), and get zero profit.

We first consider the case when θ is fixed, and study the optimal value $p^*(\theta)$ that maximizes $g(p, \theta, c)$. The main result is given in Lemma 3.

Lemma 3. *Let $\tilde{p}(\theta, c)$ be the solution to*

$$f_p(p, \theta) - c\lambda = 0, \tag{9.2}$$

and

$$c^*(\theta) = \frac{f_p(1, \theta)}{\lambda}. \tag{9.3}$$

Then

$$p^*(\theta) = \begin{cases} 1, & \text{if } c \leq c^*(\theta), \\ \\ \tilde{p}(\theta, c), & \text{if } c > c^*(\theta). \end{cases} \tag{9.4}$$

Proof. Fix a θ. Since $f(p, \theta)$ is concave over p, $f_p(p, \theta)$ is decreasing in p. Thus, $f_p(p, \theta)$ is minimized at $p = 1$.

We consider two cases.

Case (i): $c \leq c^*(\theta)$, where $c^*(\theta)$ is given in Equation 9.3.

In this case $g_p(p, \theta, c) \geq 0$. Thus, $g(p, \theta, c)$ is increasing in p for $p \in [0, 1]$. Hence, we get $p^*(\theta) = 1$.

Case (ii): $c^*(\theta) < c$. Since $f(p, \theta)$ is concave in p, then $f_p(p, \theta) - c\lambda > 0$ for $0 \leq p < \tilde{p}(\theta, c)$, and $f_p(p, \theta) - c\lambda < 0$ for $\tilde{p}(\theta, c) < p \leq 1$. Hence, $g(p, \theta, c)$ achieves its maximum at $p = \tilde{p}(\theta, c)$. Hence $p^*(\theta) = \tilde{p}(\theta, c)$. This completes the proof. □

Now we consider optimization over θ. Note that the profit function is linear in c. We have

$$g_p(p, \theta, c) = f_p(p, \theta) - c\lambda, \tag{9.5}$$

and

$$g_\theta(p, \theta, c) = f_\theta(p, \theta) - c. \tag{9.6}$$

Concavity of g in θ implies that there is a unique $\theta = \theta^*(p, c)$ that makes the derivative in Equation 9.6 zero for a given p and c, and $g(p, \theta, c)$ is maximized at $\theta = \theta^*(p, c)$.

Let

$$\tilde{c}(p, \theta) = f_\theta(p, \theta). \tag{9.7}$$

Then we see that $g_\theta(p, \theta, c) > 0$ if $c < \tilde{c}(p, \theta)$, zero if $c = \tilde{c}(p, \theta)$, and negative if $c > \tilde{c}(p, \theta)$.

Next let

$$\hat{c}(p, \theta) = \frac{f(p, \theta)}{\lambda p + \theta}. \tag{9.8}$$

Then we see that $g(p, \theta, c) > 0$ if $c < \hat{c}(p, \theta)$, zero if $c = \hat{c}(p, \theta)$, and negative if $c > \hat{c}(p, \theta)$.

Now for a fixed p, let $\theta = \bar{\theta}(p)$ be such that $\hat{c}(p, \theta) = \tilde{c}(p, \theta)$, that is,

$$(\lambda p + \theta) f_\theta(p, \theta) - f(p, \theta) = 0. \tag{9.9}$$

When $p = 0$, any θ will be the solution to Equation 9.9 since $f(0, \theta) = 0$ and $f_\theta(0, \theta) = 0$ derived from $f(p, \theta) = \lambda p m(p, \theta)$. In this case, we can choose $\theta = 0$, which implies we do not operate the system. We then show that the solution to Equation 9.9 when $p > 0$ in Lemma 4.

Lemma 4. *There is a unique non-negative solution $\theta = \bar{\theta}(p)$ to Equation 9.9 when $p > 0$.*

Proof. We know $f(p, \theta)$ is increasing concave over θ. Also, $\lim_{\theta \to 0^+} f(p, \theta) \leq 0$, and $\lim_{\theta \to \infty} f(p, \theta) = l \in (0, \infty)$. This immediately results in the unimodality of $n(p, \theta) = \frac{f(p,\theta)}{\lambda p + \theta}$. And we know

$$n_\theta(p, \theta) = \frac{(\lambda p + \theta) f_\theta(p, \theta) - f(p, \theta)}{(\lambda p + \theta)^2}.$$

Thus, there is a unique non-negative solution $\theta = \bar{\theta}(p)$ to Equation 9.9. □

Let

$$\bar{c}(p) = \hat{c}(p, \bar{\theta}(p)) = \tilde{c}(p, \bar{\theta}(p)). \tag{9.10}$$

Using Equations 9.7 and 9.8, we can write the above as

$$\bar{c}(p) = f_\theta(p, \bar{\theta}(p)) = \frac{f(p, \bar{\theta}(p))}{\lambda p + \bar{\theta}(p)}. \tag{9.11}$$

Both these formulas are useful. We see that for a fixed value of p and $c < \bar{c}(p)$, $g(p, \theta, c)$ is an increasing concave function of θ for $\theta \in [0, \bar{\theta}(p)]$ and it is maximized at $\theta = \theta^*(p, c) > \bar{\theta}(p)$. If $c = \bar{c}(p)$, $\theta^*(p, c) = \bar{\theta}(p)$ and the maximum value of $g(p, \theta, c)$, namely, $g(p, \bar{\theta}(p), c)$ is zero. If $c > \bar{c}(p)$, $\theta^*(p, c) > \bar{\theta}(p)$ and the maximum value of $g(p, \theta, c)$, namely, $g(p, \bar{\theta}(p), c)$, is negative.

We need the following two lemmas:

Lemma 5. *$\bar{c}(p)$ is an increasing function of $p \in [0, 1]$.*

Proof. Using Equation 9.8, we get

$$\begin{aligned} \hat{c}_\theta(p, \bar{\theta}(p)) &= \frac{f_\theta(p, \bar{\theta}(p))(\lambda p + \bar{\theta}(p)) - f(p, \bar{\theta}(p))}{(\lambda p + \bar{\theta}(p))^2} \\ &= 0 \text{ (using Equation 9.9)}, \end{aligned} \tag{9.12}$$

and

$$\hat{c}_p(p, \bar{\theta}(p)) = \frac{f_p(p, \bar{\theta}(p)) - \lambda f_\theta(p, \bar{\theta}(p))}{\lambda p + \bar{\theta}(p)}. \tag{9.13}$$

Then we have

$$\begin{aligned} \frac{d\hat{c}(p, \bar{\theta}(p))}{dp} &= \hat{c}_p(p, \bar{\theta}(p)) + \hat{c}_\theta(p, \bar{\theta}(p)) \frac{d\bar{\theta}(p)}{dp} \\ &= \frac{f_p(p, \bar{\theta}(p)) - \lambda f_\theta(p, \bar{\theta}(p))}{\lambda p + \bar{\theta}(p)} \\ &= \frac{\lambda(p m_p(p, \bar{\theta}(p)) + \theta m_\theta(p, \bar{\theta}(p)))}{\lambda p + \bar{\theta}(p)} \text{ (using Equation 9.9)} \\ &\geq 0 \text{ (using } p m_p(p, \bar{\theta}(p)) + \theta m_\theta(p, \bar{\theta}(p)) \geq 0). \end{aligned} \tag{9.14}$$

Thus, $\frac{\partial \bar{c}(p)}{\partial p} \geq 0$. So, $\bar{c}(p)$ is increasing in p. It is done. □

Thus,

$$\bar{c}(p) \leq \bar{c}(1). \tag{9.15}$$

Lemma 6.

$$c^*(\bar{\theta}(p)) \geq \bar{c}(p).$$

Proof.

$$\begin{aligned} c^*(\bar{\theta}(p)) - \bar{c}(p) &\geq c^*(\bar{\theta}(p)) - \bar{c}(1), \text{ (using Equation 9.15)} \\ &= \frac{f_p(1, \bar{\theta}(p))}{\lambda} - f_\theta(1, \bar{\theta}(p)), \text{ (using Equations 9.3 and 9.7)} \\ &= p m_p(p, \bar{\theta}(p)) + \theta m_\theta(p, \bar{\theta}(p)), \text{ (using Equation 9.9)} \\ &\geq 0 \text{ (using } p m_p(p, \bar{\theta}(p)) + \theta m_\theta(p, \bar{\theta}(p)) \geq 0 \text{).} \end{aligned} \tag{9.16}$$

This proves the theorem. □

With these lemmas we are ready to state the main theorem about the socially optimal policy.

Theorem 13. *Let $\bar{c}(1)$ be as given in Equation 9.11 with $p = 1$. The socially optimal operating policy (p^*, μ^*) is given by*

$$(p^*, \mu^*) = \begin{cases} (1, \lambda + \theta^*(1, c)), & \text{if } c \leq \bar{c}(1), \\ \\ (0, 0), & \text{if } c > \bar{c}(1). \end{cases}$$

Proof. Let (p, θ) be any given feasible point. From Equation 9.15, we know that $\bar{c}(p) \leq \bar{c}(1)$. Consider two cases:

Case 1: $c \leq \bar{c}(p) \leq \bar{c}(1)$.

In this case, $\theta^*(p, c) \geq \bar{\theta}(p)$. We see that $c^*(\theta)$ is increasing in θ since $f_{p,\theta}(p, \theta) \geq 0$. Then

$$\begin{aligned} c^*(\theta^*(p, c)) &\geq c^*(\bar{\theta}(p)) \\ &\geq \bar{c}(p) \text{ (based on Lemma 6)} \\ &\geq c. \end{aligned}$$

Further, based on the definition of $\theta^*(p, c)$, and Lemma 3, we have

$$g(p, \theta(p), c) \leq g(p, \theta^*(p, c), c) \leq g(1, \theta^*(p, c), c) \leq g(1, \theta^*(1, c), c).$$

Thus $g(p, \theta, c)$ is maximized at $p^* = 1$ and $\theta^* = \theta^*(1, c)$, that is, $\mu^* = \lambda + \theta^*(1, c)$.

Case 2: $\bar{c}(p) < c \leq \bar{c}(1)$. In this case $g(p,\theta) < 0$. But we know that $g(1,\theta^*(1),c) \geq 0$. Hence

$$g(p,\theta,c) \leq g(1,\theta^*(1,c),c).$$

Thus $g(p,\theta,c)$ is maximized at $p^* = 1$ and $\theta^* = \theta^*(1,c)$, that is, $\mu^* = \lambda + \theta^*(1,c)$.

Case 3: $c > \bar{c}(1)$. Then $c > \bar{c}(p)$, hence $g(p,\theta,c) < 0$ for any feasible point (p,θ). Hence the optimal profit is zero, and is obtained by setting $p^* = 0$ and $\mu^* = 0$. Thus it is not optimal to operate the system.

This proves the theorem. □

This is a surprising result. One would have expected an interior point of $(0,1)$ to arise as an optimal admission probability for some parameter values. But the above theorem says that it is either optimal to admit everybody and choose a corresponding optimal service rate, or admit no one and not operate the system at all. This is a consequence of having the flexibility of choosing both p and θ and the concavity properties of f.

The power of Theorem 13 becomes even more apparent when we consider the following Lemma.

Lemma 7. *Suppose the reward functions $f_i(p,\theta), (i = 1,2,\cdots,n)$ satisfy Assumptions 1, 2 and 3. Let*

$$f(p,\theta) = \sum_{i=1}^{n} \alpha_i f_i(p,\theta),$$

where $\alpha_i \geq 0$ for $1 \leq i \leq n$, and $\sum_{i=1}^{n} \alpha_i = 1$, $n \geq 1$. Then $f(p,\theta)$ also satisfies Assumptions 1, 2 and 3.

The proof is straightforward and is omitted. The implication of this lemma is obvious: if we know that the Theorem 13 is applicable to the reward functions $f_i(p,\theta), (i = 1,2,\cdots,n)$, then it is applicable to any convex combination of them. This can be quite useful in applications.

9.3 Decentralized Decisions

We have assumed so far that there is a central decision maker (system manager) who decides both the admission probability as well as the service rate. This leads to a rather appealing socially

optimal policy described in Theorem 13. How will the policy change if the decisions are not centralized? This will depend on who decides what and what motivates their decisions.

Recall that $m(p, \theta)$ is the revenue received by the system manager for each admitted customer. In order to decide how the customers behave if left to themselves, we need a model for how much reward a customer earns from joining the system. We shall assume that a customer gets $\alpha m(p, \theta)$ expected reward if she joins the system, where $\alpha > 0$ is a fixed constant. We assume $\alpha = 1$, without loss of generality. This situation occurs in many applications. For example, a customer may be satisfied if her service starts within a given fixed time after arrival. Suppose the system manager gets a revenue of r if the admitted customer is satisfied. The customer gets a reward of one if she is satisfied. Thus in case $\alpha = 1/r$. We always assume that if a customer is not admitted, the service manager gets no revenue. Similarly, if a customer does not join, she gets no reward. We consider three models of decentralized decision making in this section.

9.3.1 Individually Optimal Policy.

We assume the system manager is the leader who sets the service rate μ so as to maximize the net profit, but he has no control over how the customers will react to the service rate chosen by him. We further assume that the customers are followers who take this service rate μ as given and join if and only if their expected reward from joining is positive. Thus the customers behave selfishly, with no regard to the externalities they create on others. The service manager chooses the service rate taking into account this customer response. We call the resulting policy an individually optimal policy (IOP). To be precise, a policy (p_I^*, μ_I^*) is called an IOP if the profit maximizing service rate for the system manager is μ_I^* and it induces the selfish customers to join with probability p_I^*. The main result is given in the following Theorem.

Theorem 14. *The socially optimal policy (p^*, μ^*) given in Theorem 13 is also the individually optimal policy (p_I^*, μ_I^*).*

Proof. Consider the socially optimal policy (p^*, μ^*) given in Theorem 13. Suppose the service manager chooses μ^* according to this policy, without any control over the customer decisions. Then, if $c \leq \bar{c}(1)$, he uses service rate $\mu^* = \lambda + \theta^*(1, c)$; otherwise he uses $\mu^* = 0$. Clearly, in the first case the system is stable, and $m(1, \theta)$ must be positive, since otherwise it is not optimal to

operate the system. But if $m(1,\theta) > 0$, every customer will join, that is $p^* = 1$. On the other hand, in the second case, the reward from joining the system is not positive, and the customers decide not to join from selfish point of view, that is $p^* = 0$. Thus in both cases, the system manager chooses μ^* and each customer responds with p^*, that is, (p^*, μ^*) is an IOP. □

The above theorem implies that the system manager and the customer will self regulate and settle on the socially optimal policy. This regulation scheme has all the desired properties summarized by (Haviv and Oz, 2018). Firstly, each customer is free to join the queue, and if she joins the queue, she will receive the service within a finite time in our scheme. This is because the probability that the customers join the queue is one when we operate the system. When we operate the system, the queue is always stable, thus, the customers are guaranteed to receive the service within a finite time. Secondly, our queueing regime can always be made work conserving since we have not made any assumption about how the system operates. Thirdly, it is easy to see that our scheme does not have any money transfers since the customer's decision to join or not is driven by self interest. Fourthly, the policy (p^*, μ^*) says that we admit all the customers when $c \leq \bar{c}(1)$. Thus this policy is independent of the arrival rate λ. Lastly, the customers do not need to know the parameter c, their decision is based only on whether the service rate is positive or zero. The service rate decided by the service manager does depend on c.

It should be noted that the policy (p^*, μ^*) is robust since the reward function only needs to satisfy Assumptions 1, 2 and 3 under an unobservable queueing system. By contrast, the results in (Haviv and Oz, 2018) assume a linear cost reward structure for an $M/M/1$ queue in their self-regulation scheme.

9.3.2 Stackelberg Game

Now we consider a general Stackelberg game where the system manager is a leader who decides the service rate μ, and an agent for the customers is a follower who decides the joining probability p in response to μ for all customers. The Stackelberg game proceeds as follows:

The system manager first announces the service rate μ. The customer agent responds with p so as to maximize a given reward function $a(p, \mu)$. The system manager knows how the customer agent will respond, and takes this into account while setting the service rate μ to maximize his

reward function given by $a(p,\mu)+b(\mu)$. Notice the separable nature of the system manager's reward function. We make this procedure precise below.

We first define

$$p^*(\mu) = \text{argmax}\{p : a(p,\mu)\},$$

and

$$\hat{\mu} = \text{argmax}\{\mu : a(p^*(\mu),\mu) + b(\mu)\}.$$

Then

$$(p_S^*, \mu_S^*) = (p(\hat{\mu}), \hat{\mu}).$$

is called a Stackelberg solution. That is, the system manger will choose to use the service rate μ_S^* and the customer agent will respond with p_S^* under this leader/follower Stackelberg game. Now let (p^*, μ^*) be the global optimal solution of the system manager's reward function $a(\mu,p)+b(\mu)$. The main result is given in the following theorem, whose proof is almost trivial.

Theorem 15. *The globally optimal solution is a Stackelberg solution, that is,*

$$(p_S^*, \theta_S^*) = (p^*, \mu^*).$$

Proof. Suppose the system manager chooses the service rate μ^*. Then the response from the customer agent must be p^*, since

$$p^* \in \underset{\{}{\text{argmax}} f(p,\mu^*) + g(\mu^*)\} = \underset{\{}{\text{argmax}} f(p,\mu^*)\}.$$

Thus the conclusion follows. □

Now we see that our specific problem of section 9.1 is a special case of this general separable problem with

$$a(p,\mu) = f(p, \mu - \lambda p), \quad b(\mu) = -c\mu.$$

Then we have the following corollary, whose proof is omitted.

Corollary 2. *Let (p^*, μ^*) be the socially optimal policy of Theorem 13. Then it is a Satckelberg solution (p_S^*, μ_S^*).*

The above result is another indication of the robustness of the socially optimal policy, since the same policy will be followed even if decision making is decentralized between the system manager and the customer agent in a leader/follower fashion. There is no need for an external incentive to align the Stackelberg solution with the socially optimal solution.

We next consider the Nash equilibrium between the customers and service provider in the next subsection.

9.3.3 Nash Equilibrium

We again start with the general separable problem where the customer agent wants to maximize $a(p, \mu)$ and the service manager wants to maximize $a(p, \mu) + b(\mu)$. However, there is no leader/follower designation between the two. Hence we consider the Nash equalibrium which is defined precisely below. Let

$$p^*(\mu) \in \text{argmax}\{a(p, \mu)\},$$

and

$$\mu^*(p) \in \text{argmax}\{a(p, \mu) + b(\mu)\}.$$

Then (p_N^*, θ_N^*) is a Nash solution if

$$p_N^* = p^*(\mu_N^*),$$

and

$$\mu_N^* = \mu^*(p_N^*).$$

As in the previous section, let (p^*, μ^*) be the global optimal of $a(p, \mu) + b(\mu)$. Then we have following analogue of Theorem 15.

Theorem 16. *The globally optimal solution is a Nash solution, that is,*

$$(p_S^*, \theta_S^*) = (p^*, \mu^*).$$

Proof. The proof is almost the same as that of Theorem 15, and is omitted. □

Similarly, we get the following corollary:

Corollary 3. *Let (p^*, μ^*) be the socially optimal policy of Theorem 13. Then it is a Nash solution (p_N^*, μ_N^*).*

The above result is a further indication of the robustness of the socially optimal policy, since the same policy will be followed even if decision making is decentralized between the system manager and the customer agent in a symmetric fashion. There is no need for an external incentive to align the Nash solution with the socially optimal solution.

We illustrate these results by several examples in the next section.

9.4 Analytical Examples

We derive the socially optimal policy under four different settings in the examples below. From the results in the previous section, we know that this is also the individually optimal policy, the Stckelberg policy and the Nash policy.

Example 1. Queueing Time Dependent Binary Reward in $M/M/1$ Queue. Suppose the service times in the system are iid $\exp(\mu)$ random variables. Then the system is an $M/M/1$ system with arrival rate λp and service rate μ. It is stable if $\mu > \lambda p$. Suppose the customer receives one dollar if her queueing time (that is the time spent in the system until the service starts) is b or less, where $b \geq 0$ is a given constant. Then the reward rate from each admitted customer is given by

$$m(p, \theta) = 1 - \frac{\lambda p e^{-b\theta}}{\lambda p + \theta}, \tag{9.17}$$

and hence

$$f(p, \theta) = \lambda p (1 - \frac{\lambda p e^{-b\theta}}{\lambda p + \theta}), \;\; 0 \leq p \leq 1, \theta \geq 0, \tag{9.18}$$

with

$$f_p(p, \theta) = \frac{\lambda^3 (1 - e^{-\theta b}) p^2 + 2\theta\lambda^2 (1 - e^{-\theta b}) p + \lambda\theta^2}{(\theta + \lambda p)^2}, \tag{9.19}$$

and

$$f_\theta(p, \theta) = \frac{\lambda^2 p^2 e^{-\theta b}}{(\theta + \lambda p)^2} + \frac{b\lambda^2 p^2 e^{-\theta b}}{\theta + \lambda p} \geq 0. \tag{9.20}$$

It is straightforward to verify that assumptions 1 and 2 are satisfied. We have

$$pm_p(p,\theta) + \theta m_\theta(p,\theta) = \frac{b\lambda p\theta e^{-b\theta}}{\lambda p + \theta} \geq 0,$$

which verifies assumption 3.

When $p = 1$, $\theta = \theta^*(1, c)$ maximizes the profit $g(1, \theta, c)$. It satisfies the equation $f_\theta(1, \theta) = c$, which reduces to

$$\frac{\lambda^2 e^{-\theta b}}{(\theta + \lambda)^2} + \frac{b\lambda^2 e^{-\theta b}}{\theta + \lambda} = c. \tag{9.21}$$

For $p = 1$, Equation 9.9 reduces to

$$\frac{2\lambda e^{-\theta b}}{\lambda + \theta} + \lambda b e^{-\theta b} = 1. \tag{9.22}$$

Let $\bar{\theta}(1)$ be the unique solution to the above equation. Using the first equality in Equation 9.11, we see that the quantity $\bar{c}(1)$, defined in Equation 9.11 with $p = 1$, is given by

$$\bar{c}(1) = \frac{\lambda^2 e^{-\bar{\theta}(1)b}}{(\bar{\theta}(1) + \lambda)^2} + \frac{b\lambda^2 e^{-\bar{\theta}(1)b}}{\bar{\theta}(1) + \lambda}. \tag{9.23}$$

Then Theorem 13 says that if $c \leq \bar{c}(1)$, the profit is maximized at $p^* = 1$, and $\mu^* = \lambda + \theta^*(1, c)$. If $c > \bar{c}(1)$, the profit is maximized at $p^* = 0$, and $\mu^* = 0$.

Example 2. Waiting Time Dependent Binary Reward in $M/M/1$ Queue. Consider the setting of Example 1. Suppose the customer receives a reward of one dollar if her waiting time (defined as queuing time plus service time) is b or less, where $b \geq 0$ is a given constant.

Then the expected reward from an admitted customer is given by

$$m(p, \theta) = 1 - e^{-\theta b}, \tag{9.24}$$

and hence the reward rate from the admitted customers is given by

$$f(p, \theta) = \lambda p(1 - e^{-\theta b}), \quad 0 \leq p \leq 1, \theta \geq 0, \tag{9.25}$$

with

$$f_p(p,\theta) = \lambda(1 - e^{-\theta b}), \tag{9.26}$$

and

$$f_\theta(p,\theta) = \lambda p b e^{-\theta b} \geq 0. \tag{9.27}$$

It is straightforward to verify that assumptions 1 and 2 are satisfied. We have

$$p m_p(p,\theta) + \theta m_\theta(p,\theta) = \theta b e^{-\theta b} \geq 0,$$

which verifies assumption 3.

When $p = 1$, $\theta = \theta^*(1,\theta)$ maximizes the profit $g(1,\theta,c)$. It satisfies the equation $f_\theta(1,\theta) = c$, which reduces to

$$\lambda b e^{-\theta b} = c. \tag{9.28}$$

For $p = 1$, Equation 9.9 reduces to

$$(\lambda + \theta) b e^{-\theta b} - (1 - e^{-\theta b}) = 0. \tag{9.29}$$

Let $\bar{\theta}(1)$ be the unique solution to the above equation. Using the first equality in Equation 9.11, we see that the quantity $\bar{c}(1)$, defined in Equation 9.11 with $p = 1$, is given by

$$\bar{c}(1) = \lambda b e^{-\bar{\theta}(1) b}. \tag{9.30}$$

Then Theorem 13 says that if $c \leq \bar{c}(1)$, the profit is maximized at $p^* = 1$, and $\mu^* = \lambda + \theta^*(1,c)$. If $c > \bar{c}(1)$, the profit is maximized at $p^* = 0$, and $\mu^* = 0$.

Example 3. Queueing Time Related Holding Cost for $M/G/1$ Queue. We consider an $M/G/1$ queue with arrival rate λp, and a general service distribution. Let S denote a random service time with a general service distribution $G(\cdot)$ with $C_B^2 = \frac{Var(s)}{E^2(S)}$ being a constant. C_B^2 is the squared coefficient of variation of the service distribution. Note that when $C_B^2 = 1$, the queueing system will be reduced to an $M/M/1$ Queue. Assume the average service rate $\mu = \frac{1}{E(S)} > \lambda p$ to make sure the queue is stable. Suppose each customer receives r dollars if she joins the queue, but

it costs h dollars per unit time to stay in the queue. Based on (Shortle et al., 2018), the expected queueing time of the customer is given by

$$E(W) = \frac{1 + C_B^2}{2} \frac{\lambda p}{\theta(\lambda p + \theta)} = \frac{\lambda p k}{\theta(\lambda p + \theta)},$$

where $k = \frac{1+C_B^2}{2}$.

The revenue function is given by:

$$f(p, \theta) = \lambda p \left(r - \frac{\lambda p k h}{\theta(\lambda p + \theta)} \right), \quad 0 \leq p \leq 1, \theta > 0, \tag{9.31}$$

with

$$f_p(p, \theta) = \lambda r - \frac{hk\lambda^2 p(2\theta + \lambda p)}{\theta(\lambda p + \theta)^2}, \tag{9.32}$$

and

$$f_\theta(p, \theta) = \frac{hk\lambda^2 p^2(2\theta + \lambda p)}{\theta^2(\theta + \lambda p)^2} \geq 0. \tag{9.33}$$

It is straightforward to verify that assumptions 1 and 2 are satisfied. We have

$$p m_p(p, \theta) + \theta m_\theta(p, \theta) = \frac{hk\lambda p}{\theta(\theta + \lambda p)} \geq 0,$$

which verifies assumption 3.

When $p = 1$, $\theta = \theta^*(1, \theta)$ maximizes the profit $g(1, \theta, c)$. It satisfies the equation $f_\theta(1, \theta) = c$, which reduces to

$$\frac{hk\lambda^2(2\theta + \lambda)}{\theta^2(\theta + \lambda)^2} = c. \tag{9.34}$$

For $p = 1$, Equation 9.9 reduces to

$$\frac{hk\lambda(2\theta + \lambda)}{\theta^2(\theta + \lambda)} - \left(r - \frac{h\lambda k}{\theta(\lambda + \theta)} \right) = 0. \tag{9.35}$$

Let $\bar{\theta}(1)$ be the unique solution to the above equation. Using the first equality in Equation 9.11, we see that the quantity $\bar{c}(1)$, defined in Equation 9.11 with $p = 1$, is given by

$$\bar{c}(1) = \frac{hk\lambda^2(2\bar{\theta}(1) + \lambda)}{(\bar{\theta}(1))^2(\lambda + \bar{\theta}(1))^2}. \tag{9.36}$$

Then Theorem 13 says that if $c \leq \bar{c}(1)$, the profit is maximized at $p^* = 1$, and $\mu^* = \lambda + \theta^*(1, c)$. If $c > \bar{c}(1)$, the profit is maximized at $p^* = 0$, and $\mu^* = 0$.

Example 4. Waiting Time Related Holding Cost for $M/G/1$ Queue. We consider the same setting in Example 3. Suppose each customer receives r dollars if she joins the queue, but it costs h dollars per unit time to stay in the system. We consider an $M/G/1$ queue with arrival rate λp, and a general service distribution with the average service rate $\mu > \lambda p$. Based on (Shortle et al., 2018), the expected waiting time of the customer is given by

$$E(W) = \frac{1 + C_B^2}{2} \frac{\lambda p}{\theta(\lambda p + \theta)} + \frac{1}{\lambda p + \theta} = \frac{\lambda p k + \theta}{\theta(\lambda p + \theta)}.$$

The revenue function is given by:

$$f(p, \theta) = \lambda p \left(r - \frac{h(\lambda p k + \theta)}{\theta(\lambda p + \theta)} \right), \quad 0 \leq p \leq 1, \theta > 0, \tag{9.37}$$

with

$$f_p(p, \theta) = \lambda r - \frac{h\lambda(k\lambda^2 p^2 + 2k\lambda\theta p + \theta^2)}{\theta(\lambda p + \theta)^2}, \tag{9.38}$$

and

$$f_\theta(p, \theta) = \frac{h\lambda p(k\lambda^2 p^2 + 2k\lambda\theta p + \theta^2)}{\theta^2(\lambda p + \theta)^2} \geq 0. \tag{9.39}$$

It is straightforward to verify that assumptions 1 and 2 are satisfied. We have

$$p m_p(p, \theta) + \theta m_\theta(p, \theta) = \frac{h(k\lambda p + \theta)}{\theta(\theta + \lambda p)} \geq 0,$$

which verifies assumption 3.

When $p = 1$, $\theta = \theta^*(1, \theta)$ maximizes the profit $g(1, \theta)$. It satisfies the equation $f_\theta(1, \theta) = c$, which reduces to

$$\frac{h\lambda(k\lambda^2 + 2k\lambda\theta + \theta^2)}{\theta^2(\lambda + \theta)^2} = c. \tag{9.40}$$

For $p = 1$, Equation 9.9 reduces to

$$\frac{h(k\lambda^2 + 2k\lambda\theta + \theta^2)}{\theta^2(\lambda + \theta)} - \left(r - \frac{h(\lambda k + \theta)}{\theta(\lambda + \theta)} \right) = 0. \tag{9.41}$$

Let $\bar{\theta}(1)$ be the unique solution to the above equation. Using the first equality in Equation 9.11, we see that the quantity $\bar{c}(1)$, defined in Equation 9.11 with $p = 1$, is given by

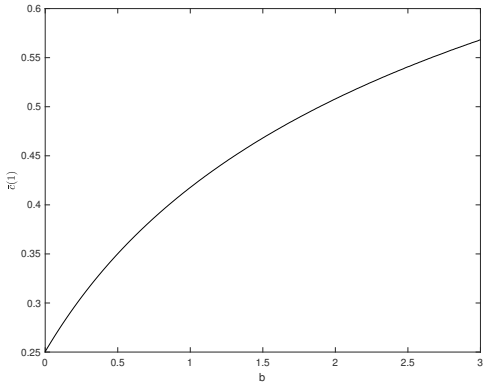
$$\bar{c}(1) = \frac{h\lambda(k\lambda^2 + 2k\lambda\bar{\theta}(1) + (\bar{\theta}(1))^2)}{(\bar{\theta}(1))^2(\lambda + \bar{\theta}(1))^2}. \tag{9.42}$$

Then Theorem 13 says that if $c \leq \bar{c}(1)$, the profit is maximized at $p^* = 1$, and $\mu^* = \lambda + \theta^*(1, c)$. If $c > \bar{c}(1)$, the profit is maximized at $p^* = 0$, and $\mu^* = 0$.

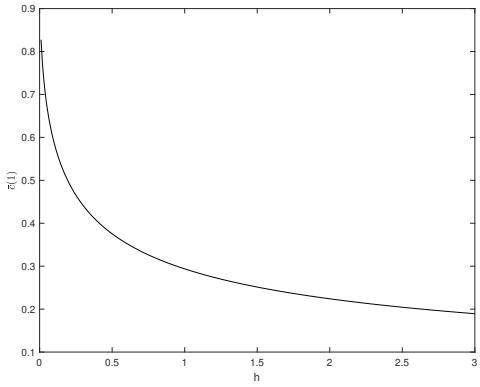
9.5 Numerical Results

In this section, we present the numerical results for the joint admission and service rate control problem. Specifically, we first consider the binary revenue structure and holding cost reward structure on the queueing time in examples 1 and 3, respectively. Here, we use holding cost reward structure and linear reward structure interchangeably. We set $\lambda = 1$, $r = 1$ and $C_B^2 = 1$ without loss of generality, and study the behavior of $\bar{c}(1)$ as a function of b, μ^* and $g(p^*, \mu^*, c)$ as a function of c and b in example 1, and study the behavior of μ^* and $g(p^*, \mu^*, c)$ as a function of c and h in example 3.

We first plot $\bar{c}(1)$ as a function of $b \in [0, 3]$ and $h \in (0, 3]$ in Figures 9.1a and 9.1b, respectively. It should be noted that c does not play any role in computing $\bar{c}(1)$. We can see that $\bar{c}(1)$ increases with b and decreases with h since the reward is increasing in b, and decreasing in h in examples 1 and 3, respectively. This is as expected. It is also interesting to see that $\bar{c}(1)$ is concave in b, but convex in h. It implies the increasing rate of $\bar{c}(1)$ will become smaller with the increase of b, and the decreasing rate of $\bar{c}(1)$ will become smaller with the increase of h. This is because the binary reward structure computes the reward over b in a cumulative way, but the holding cost reward structure computes the reward over h in a marginal way. This is consistent with the law of diminishing marginal utility. By applying the results of Theorem 13, we see that $\bar{c}(1)$ shows the maximum value of c that will bring a profitable operation. This can be called the server value. Thus when $b = 1$ in example 1, $\bar{c}(1)$ is equal to .4177. Thus using Theorem 13, we see that if $c > .4177$, $p^* = 0$, and $\mu^* = 0$ in example 1. Otherwise, $p^* = 1$, and $\mu^* = \lambda + \theta^*(1, c)$. Similarly, in example 3, when $h = 1$ in example 1, $\bar{c}(1)$ is equal to .2938, and similar result can be derived.

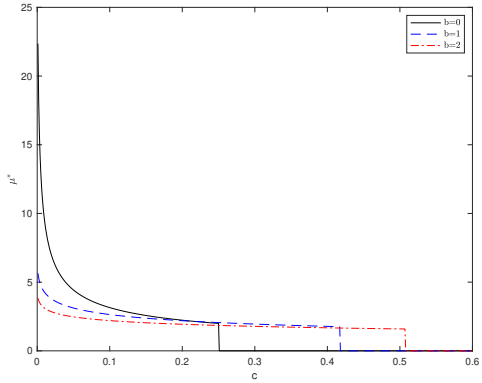


(a) $\bar{c}(1)$ as a function of b in example 1

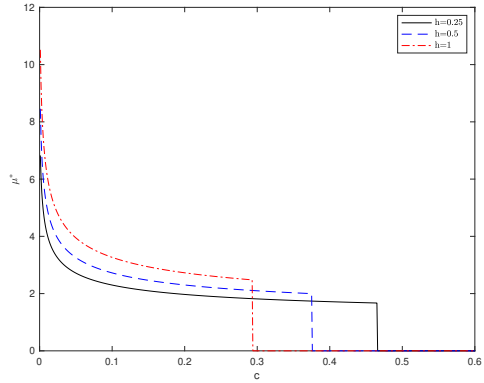


(b) $\bar{c}(1)$ as a function of h in example 3

Figure 9.1: $\bar{c}(1)$ as a function of b and h in examples 1 and 3, respectively



(a) μ^* as a function of c in example 1

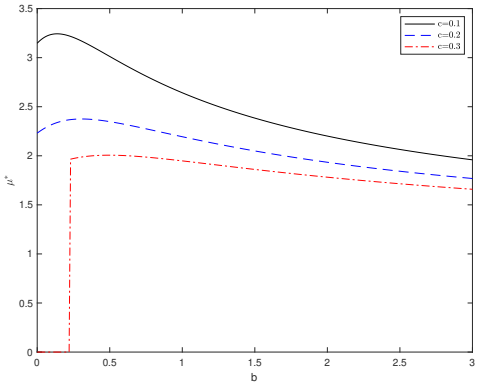


(b) μ^* as a function of c in example 3

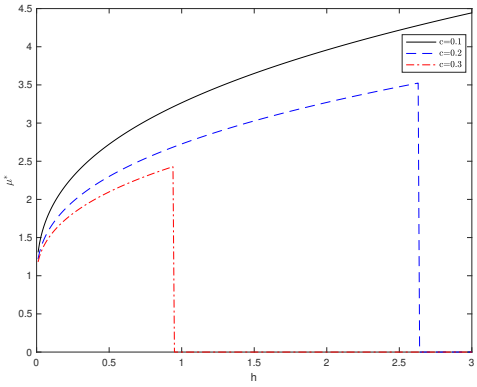
Figure 9.2: μ^* as a function of c in examples 1 and 3, respectively

We next show μ^* as a function of $c \in [0, 0.6]$ for $b = 0, 1, 2$ in Figure 9.2a, and for $h = .25, .5, 1$ in Figure 9.2b. We can see that μ^* is decreasing convex in c. This implies if the per unit service cost becomes larger, it will induce the service provider to use less service rate. The convexity in c is due to the law of diminishing marginal utility over c. It can also be observed that μ^* will become 0 once it is beyond $\bar{c}(1)$. This is consistent with the result obtained in the previous paragraph.

We then show μ^* as a function of $b \in [0, 3]$ for $c = .1, .2, .3$ in Figure 9.3a, and as a function of $h \in [0, 3]$ for $c = .1, .2, .3$ in Figure 9.3b. In Figure 9.3a, we see that μ^* is first increasing concave in b, and then decreasing in b. This is very surprising especially when it is compared with the concavity of μ^* over b under the holding cost reward structure, which will be shown later. It shows when b is small, namely, when it is very difficult to earn the reward, μ^* is increasing in b. It implies that the increase in service rate in increasing the reward is more important in increasing the profit



(a) μ^* as a function of b in example 1



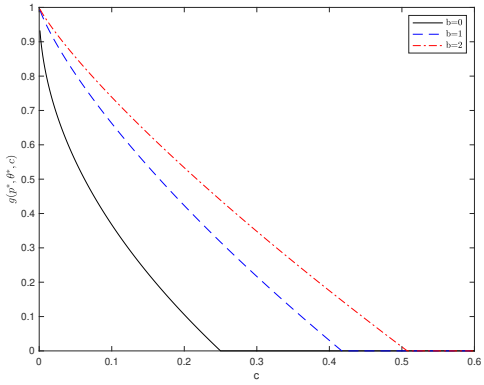
(b) μ^* as a function of h in example 3

Figure 9.3: μ^* as a function of b and h in examples 1 and 3, respectively

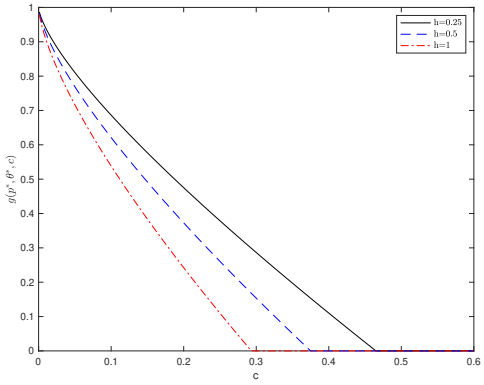
compared to the cost incurred by the increased service rate. This effect is decreasing in b, and once b is beyond a critical value, with the further increase of b, μ^* is decreasing in b. It implies the service provider can use less service rate to make more profit instead of using more service rate when b is relatively larger, namely, when it becomes easier to earn the reward. By contrast, in Figure 9.3b, we see that μ^* is always increasing concave over h until it drops down to zero. With the increase of h, it becomes more difficult to earn the reward. In this case, the service provider tends to use a larger service rate to earn more reward, which outweighs the cost incurred by the server. Thus, he can get more profit. Due to the concavity, we see that the effect is decreasing in h.

Finally, we show the optimal profit $g(p^*, \mu^*, c)$ as a function of $c \in [0, .6]$ for $b = 0, 1, 2$, in Figure 9.4a, and as a function of $c \in [0, .6]$ for $h = .25, .5, 1$, in Figure 9.4b. $g(p^*, \mu^*, c)$ is decreasing convex in c since it becomes more expensive to increase the service rate with the increase of c. With the increase of c, the decreasing rate of $g(p^*, \mu^*, c)$ will become smaller due to the law of diminishing marginal utility over c. In Figure 9.5, we see that $g(p^*, \mu^*, c)$ is increasing concave in b, and decreasing convex in h since the reward becomes larger with the increase of b or the decrease of h. The concavity over b and the convexity over h can be similarly explained using the argument on μ^*.

Further, we consider the binary revenue structure and holding cost reward structure on the waiting time in examples 2 and 4, respectively. We show the detailed figures as follows. We first plot $\bar{c}(1)$ as a function of $b \in [0, 3]$ and $h \in (0, 3]$ in Figures 9.6a and 9.6b, respectively.

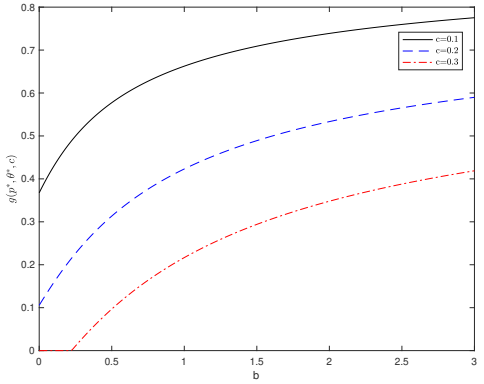


(a) Optimal profit as a function of c in example 1

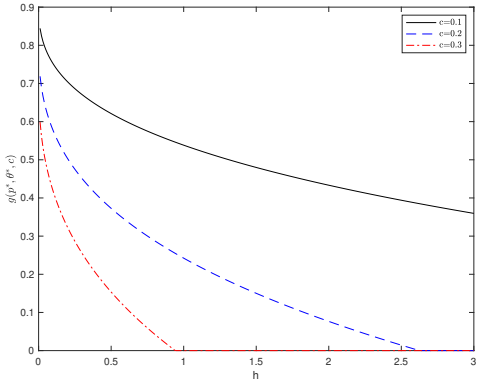


(b) Optimal profit as a function of c in example 3

Figure 9.4: Optimal profit as a function of c in examples 1 and 3, respectively

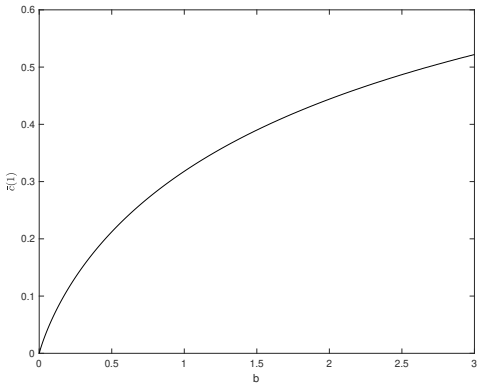


(a) Optimal profit as a function of b in example 1

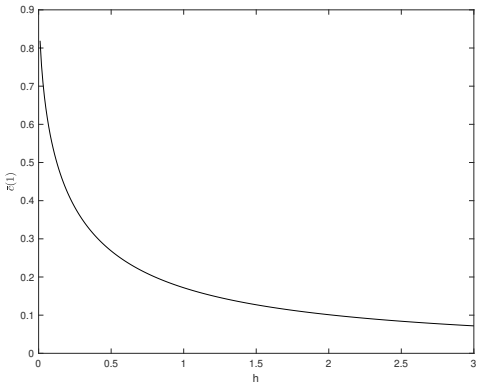


(b) Optimal profit as a function of h in example 3

Figure 9.5: Optimal profit as a function of b and h in examples 1 and 3, respectively

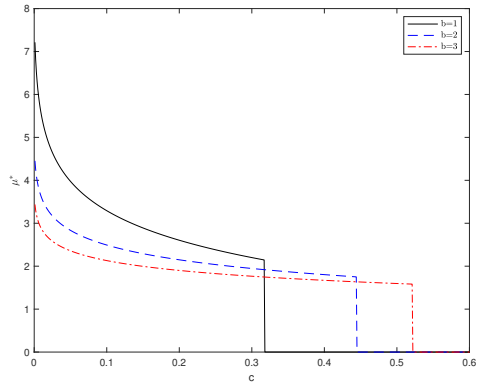


(a) $\bar{c}(1)$ as a function of b in example 2

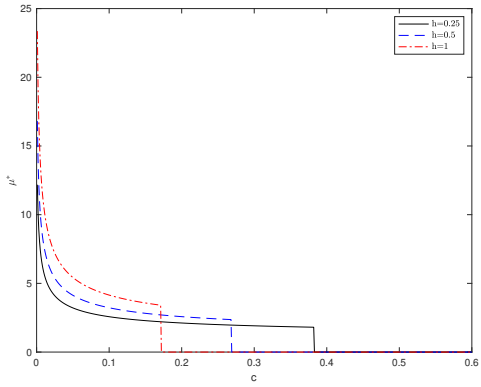


(b) $\bar{c}(1)$ as a function of h in example 4

Figure 9.6: $\bar{c}(1)$ as a function of b and h in examples 2 and 4, respectively

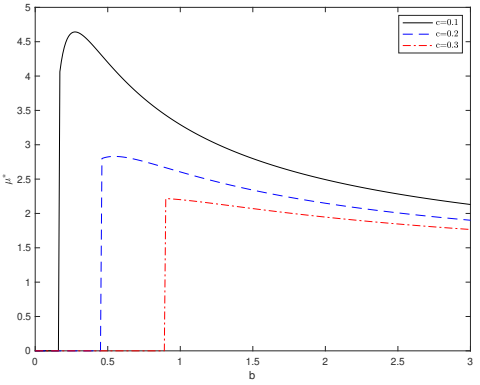


(a) μ^* as a function of c in example 2

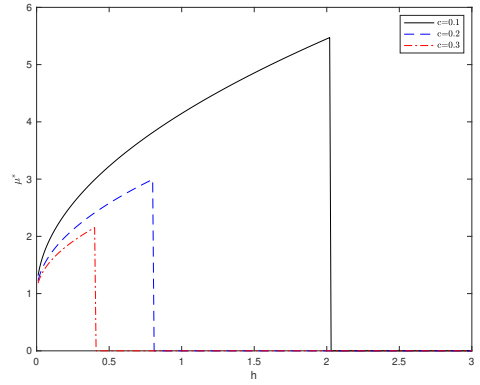


(b) μ^* as a function of c in example 4

Figure 9.7: μ^* as a function of c in examples 2 and 4, respectively



(a) μ^* as a function of b in example 2



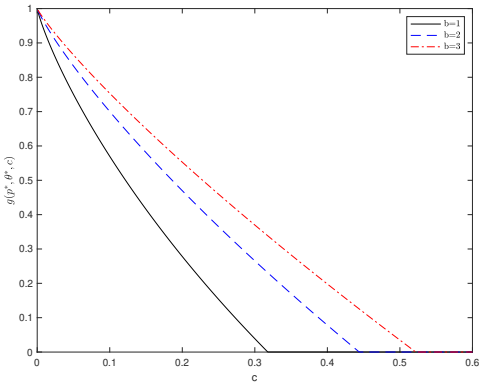
(b) μ^* as a function of h in example 4

Figure 9.8: μ^* as a function of b and h in examples 2 and 4, respectively

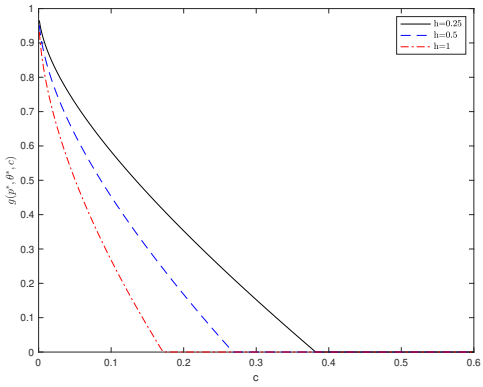
We next show μ^* as a function of $c \in [0, 0.6]$ for $b = 1, 2, 3$ in Figure 9.7a, and for $h = .25, .5, 1$ in Figure 9.7b.

We then show μ^* as a function of $b \in [0, 3]$ for $c = .1, .2, .3$ in Figure 9.8a, and as a function of $h \in [0, 3]$ for $c = .1, .2, .3$ in Figure 9.8b.

Finally, we show the optimal profit $g(p^*, \mu^*, c)$ as a function of $c \in [0, .6]$ for $b = 1, 2, 3$, in Figure 9.9a, and as a function of $c \in [0, .6]$ for $h = .25, .5, 1$, in Figure 9.9b. Then we show the optimal profit $g(p^*, \mu^*, c)$ as a function of $b \in [0, 3]$ for $c = .1, .2, .3$, in Figure 9.10a, and as a function of $h \in [0, 3]$ for $c = .1, .2, .3$, in Figure 9.10b. We can see that the monotonicity and concavity of server value, optimal service rate and optimal profit in examples 2 and 4 follow the similar pattern to those in examples 1 and 3, respectively. It implies the queueing time and waiting time reward structures follow the similar pattern in monotonicity and concavity. It further shows

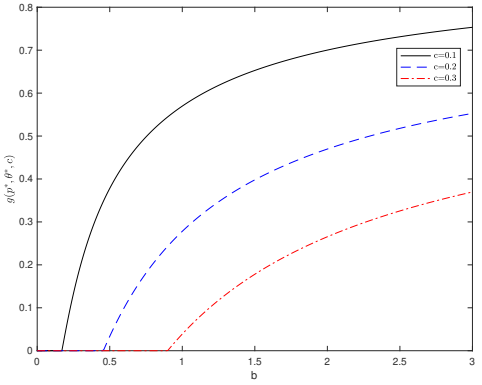


(a) Optimal profit as a function of c in example 2

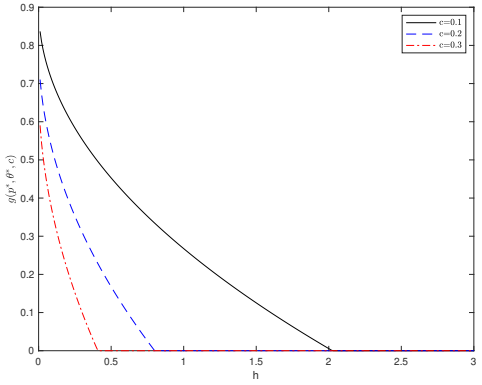


(b) Optimal profit as a function of c in example 4

Figure 9.9: Optimal profit as a function of c in examples 2 and 4, respectively



(a) Optimal profit as a function of b in example 2



(b) Optimal profit as a function of h in example 4

Figure 9.10: Optimal profit as a function of b and h in examples 2 and 4, respectively

the robustness of our proposed optimal joint admission and service rate control policy. No matter whether the system manager cares about the queueing time or the waiting time, they can use similar pattern to direct their practice. This would make our proposed policy much more applicable to real operations.

9.6 Conclusions

In this project, we consider the joint admission and service rate control problem for an unobservable single server queueing system. We first introduce the general reward structure and the conditions the general reward structure should satisfy. Then we show the detailed steps to prove the optimal joint admission and service rate control policy under the centralized decision case. Fur-

ther, we analyze the joint policy under the decentralized decision case, namely, the decentralized Stackelberg game, two-player Stackelberg game and two-player Nash game. Then we give several analytical examples under both binary and linear reward structures. Finally, we do extensive numerical analysis on the server value, optimal service rate, and optimal profit under different reward structures.

We show that it is optimal to admit all the customers when the per unit service cost is less than or equal to a critical level, called the server value, otherwise, it is optimal to admit no one. This optimal policy works for any reward structure as long as it satisfies the assumptions specified in Section 9.1. This makes the joint policy very easy to implement, and much more applicable to real operations. This policy would also make the customers behave in a socially optimal way with self-regulation. It not only has the desired properties proposed by previous research, but also has the additional properties including the robustness to reward structure and queueing system. We also show that the centralized decision case, Stackelberg equilibrium and Nash equilibrium are equivalent. This further enhances the desirability of our proposed policy. In the analytical examples, we show the detailed steps to compute the joint admission and service rate policy analytically. In the numerical analysis, we find a surprising result that the optimal service rate first increases and then decreases in the service level parameters under binary reward structure. This is unexpected. However, it is always increasing in the holding cost per unit time until it drops down to zero under linear reward structure. This manifests the importance and the complexity of the joint admission and service rate control problem.

APPENDIX A

RATIONALE TO CHOOSE SCHEDULED ARRIVAL TIME, ORIGIN AIRPORT AND AIRCRAFT TYPE AS COVARIATES

In this section, we show the reason why we finally choose scheduled arrival time, origin airport and aircraft type as covariates in the first project in the first part. Based on the research from (Deshpande and Arıkan, 2012), they consider seven types of covariates: (1) route, (2) carrier, (3) origin airport, (4) destination airport, (5) congestion at the origin airport, (6) congestion at the destination airport, and (7) aircraft-specific variable. We focus on the aircraft assignment problem operated by Delta Airlines at Atlanta airport (destination airport). So it means the destination airport and carrier can also be regarded as the covariates we consider. We capture the aircraft-specific variable by considering the aircraft type. We capture congestion at the destination airport by considering scheduled arrival time at the destination airport. Since we only consider the assignment at Atlanta airport without considering the route, it makes no sense to consider route as a covariate. The only left covariate is the congestion at the origin airport. It is possible to consider scheduled departure time at the origin airport of the flight to capture the congestion at the origin airport. However, it would have a similar effect compared to that of the scheduled arrival time at the destination airport. And if we further consider the scheduled departure time as another covariate, it would make the number of flights in each cell too sparse. It implies if we consider all the covariates of arrival delay, then we may only have a few flight observations in the data to compute each element of the cost matrix. If we consolidate all flights in the data to estimate the arrival delay distribution, then we would not be able to consider the effect from the covariates on the arrival delay when computing the cost matrix. There is a balance in choosing the right number of covariates. Thus we finally choose the scheduled arrival time, origin airport and aircraft type as covariates. Then we further show why these three covariates are important as follows.

We first fix the origin airport as Chicago airport and fix the aircraft type as MD-88/MD-90-30, and then draw the empirical cumulative distribution of arrival delay of the incoming flights arriving in $[14, 15)$ and $[20, 21)$, respectively. It is shown in Figure A.1. From the figure, we can also see that there is a big difference in arrival delay distribution between intervals $[14, 15)$ and $[20, 21)$.

We further fix the scheduled arrival time in interval $[9, 10)$ and fix aircraft type as MD-88/MD-90-30, and then draw the empirical cumulative distribution of arrival delay of the incoming flights

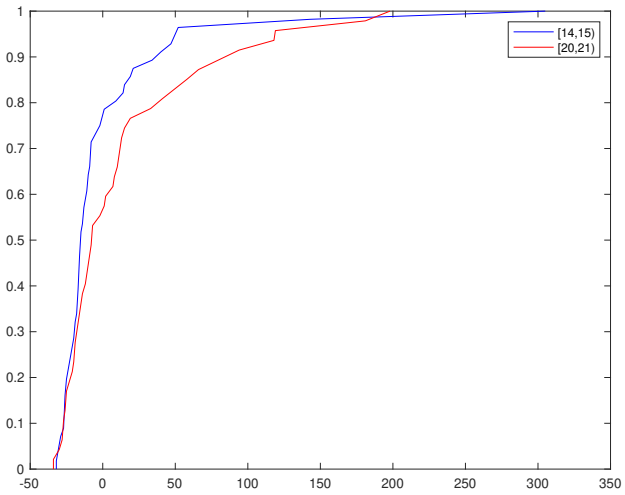


Figure A.1: Comparison on the Empirical Cumulative Distribution of Arrival Delay between Intervals $[14, 15)$ and $[20, 21)$ with the Origin Airport being Chicago Airport and the Aircraft Type being MD-88/MD-90-30

from Columbia airport and Chicago airport, respectively. It is shown in Figure A.2. From the figure, we can also see that there is a big difference in arrival delay distribution between Columbia airport and Chicago airport.

We finally fix the scheduled arrival time in interval $[8, 9)$ and fix origin airport as Chicago airport, and then draw the empirical cumulative distribution of arrival delay of the incoming flights belonging to Boeing 737-932ER and MD-88/MD-90-30, respectively. It is shown in Figure A.3. From the figure, we can see that there is a big difference in arrival delay distribution between Boeing 737-932ER and MD-88/MD-90-30.

We can see that the three covariates including the scheduled arrival time, origin airport and aircraft type have an important effect on the arrival delay distribution. So, we finally choose the scheduled arrival time, origin airport and aircraft type as covariates.

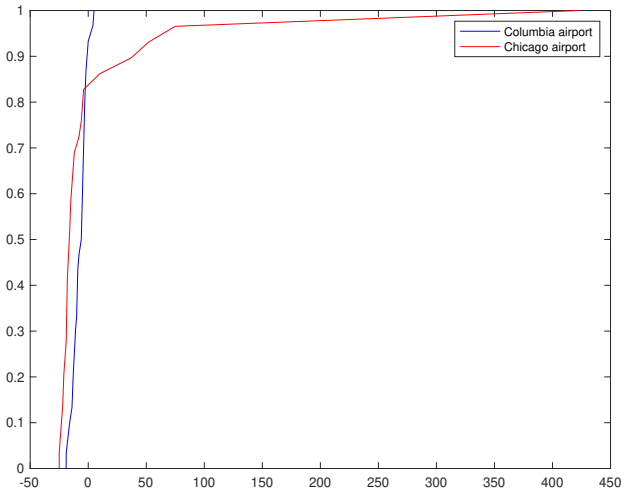


Figure A.2: Comparison on the Empirical Cumulative Distribution of Arrival Delay between Columbia Airport and Chicago Airport with Scheduled Arrival Time in $[9, 10)$ and Aircraft Type being MD-88/MD-90-30

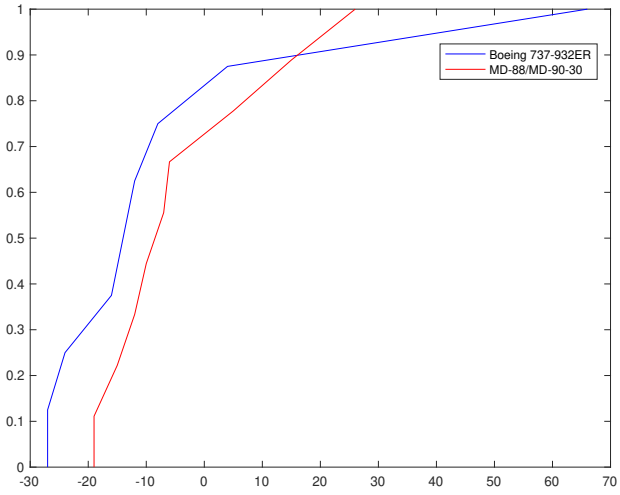


Figure A.3: Comparison on the Empirical Cumulative Distribution of Arrival Delay between Boeing 737-932ER and MD-88/MD-90-30 with Scheduled Arrival Time in $[8, 9)$ and Origin Airport being Chicago Airport

APPENDIX B

CLUSTER LABEL FOR EACH FLIGHT

Table B.1: Cluster Label for Each flight in N_1

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
27	JFK	IAD	9:45	11:05	0	10	22	23	36	44	53	405	1
32	CLT	BOS	9:55	12:15	0	0	0	0	4	14	14	332	1
48	PWM	JFK	6:30	8:00	0	0	0	0	0	0	86	429	1
50	RIC	BOS	12:10	13:43	2	3	5	8	14	19	64	372	1
76	JFK	PIT	15:40	17:47	0	0	0	8	25	40	40	399	1
77	JFK	PIT	7:05	8:55	0	0	0	0	0	16	58	366	1
21	ORD	JFK	17:55	21:35	0	6	15	16	17	154	196	202	2
22	PBI	BOS	16:05	19:10	0	2	7	22	28	68	134	191	2
33	JFK	CLT	11:10	13:04	0	5	12	19	26	57	192	218	2
59	JFK	CMH	11:15	13:05	0	0	0	4	26	55	155	172	2
60	JFK	CMH	7:20	9:20	0	0	0	0	3	6	204	226	2
10	BOS	JFK	15:50	17:07	38	44	47	64	88	147	161	249	3
12	CLT	JFK	13:45	15:45	0	0	1	17	108	132	163	273	3
36	PIT	JFK	14:35	16:10	0	20	54	94	150	181	238	251	3
79	JFK	ORD	15:25	17:14	31	39	86	93	122	170	205	229	3
101	AUS	JFK	10:40	15:28	15	25	28	49	106	146	190	207	3
3	BOS	JFK	18:55	20:13	14	22	23	32	56	58	81	145	4
16	CMH	JFK	13:40	15:32	7	22	35	36	75	103	136	177	4
37	PIT	BOS	13:25	15:07	26	31	32	59	63	67	102	163	4
54	RDU	JFK	19:40	21:15	19	29	36	39	57	70	110	176	4
57	RDU	BOS	16:05	18:15	19	20	23	52	62	64	137	150	4
61	PWM	JFK	18:20	19:35	34	37	47	49	62	73	105	162	4
64	JFK	AUS	19:45	23:06	0	31	32	44	51	78	89	126	4
70	JFK	BOS	16:15	17:56	16	24	29	41	43	82	83	191	4
72	JFK	PBI	19:15	22:30	26	28	31	43	46	49	68	158	4
73	JFK	BOS	17:40	19:21	29	32	33	50	61	74	79	192	4
78	JFK	BOS	8:45	10:00	35	36	46	48	53	67	71	218	4
93	HOU	JFK	10:55	15:22	26	39	41	41	54	95	136	203	4
9	BOS	JFK	10:35	11:47	0	2	4	4	9	20	24	25	5
14	CLT	JFK	8:35	10:36	0	0	4	8	8	10	30	80	5
15	BOS	PIT	12:15	14:02	0	0	0	0	0	0	0	38	5
18	BOS	PBI	12:15	15:30	0	0	0	2	14	15	18	19	5
19	CMH	BOS	10:55	12:39	5	5	12	14	16	23	30	54	5
20	CMH	JFK	10:00	11:42	0	0	0	0	0	12	21	36	5
23	JFK	IAD	21:50	23:16	3	4	11	12	16	17	31	35	5

Continued on next page

Table B.1 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
24	PBI	JFK	5:25	8:00	0	0	13	16	18	20	28	29	5
29	BOS	TPA	14:35	17:55	0	0	4	6	10	15	44	56	5
34	JFK	CLT	6:00	8:01	0	0	0	0	9	12	17	85	5
40	BOS	RIC	9:55	11:37	0	0	0	0	0	4	19	26	5
42	BOS	RIC	19:45	21:30	0	0	0	0	0	0	0	1	5
43	BUF	BOS	10:20	11:40	0	0	0	0	0	0	0	5	5
56	RDU	JFK	5:05	6:37	0	0	0	0	0	0	11	17	5
67	JFK	PWM	15:55	17:40	7	8	10	15	17	34	41	65	5
92	JFK	RDU	20:40	22:52	5	18	19	20	24	28	33	61	5
95	BOS	BUF	8:00	9:36	0	0	0	0	0	0	0	53	5
96	BOS	AUS	6:30	9:50	0	0	0	0	0	0	27	35	5
97	BOS	CMH	8:00	10:20	0	0	0	0	0	0	0	113	5
99	BOS	IAD	8:00	9:40	0	0	0	0	0	0	8	81	5
100	ACK	JFK	18:55	20:30	0	0	0	3	3	27	35	42	5
105	BNA	JFK	5:05	8:25	0	0	0	0	0	0	12	16	5
106	BNA	JFK	10:20	13:39	0	0	0	0	10	11	21	70	5
4	JFK	HOU	7:10	10:14	16	18	19	20	27	34	55	215	6
25	JFK	IAD	16:25	18:13	0	15	16	26	28	43	55	194	6
31	SYR	JFK	13:00	14:10	4	4	6	20	27	32	57	189	6
46	JFK	AUS	8:05	11:14	0	0	0	0	0	10	83	204	6
49	RIC	BOS	6:00	7:30	0	0	0	4	8	21	25	201	6
52	RDU	JFK	11:30	13:05	0	0	0	0	0	9	20	205	6
53	JFK	CLT	13:40	15:39	16	17	28	29	29	32	46	242	6
58	RIC	JFK	18:45	20:07	15	16	23	26	28	38	45	176	6
66	JFK	BNA	8:05	9:40	17	25	26	31	37	42	55	211	6
69	IAD	BOS	7:50	9:22	0	0	0	15	26	50	51	239	6
71	IAD	BOS	19:50	21:20	6	10	10	24	29	39	41	191	6
87	JFK	RIC	11:30	12:57	0	0	0	0	7	20	48	237	6
98	BOS	CLT	7:10	9:20	19	19	24	32	35	37	53	176	6
1	BOS	JFK	17:40	19:03	42	47	68	83	88	99	143	153	7
30	IAD	MCO	17:20	19:45	18	54	61	91	95	120	125	176	7
82	IAD	JFK	16:25	17:53	53	75	81	94	96	126	167	201	7
83	IAD	JFK	19:00	20:25	45	47	55	60	93	95	103	154	7
91	HOU	JFK	16:00	20:38	41	56	80	116	118	130	136	153	7
2	JFK	HOU	12:20	15:20	0	7	17	19	20	26	29	81	8
5	BOS	IAD	15:00	16:46	15	18	18	40	41	54	65	77	8
6	JFK	SYR	11:10	12:22	0	1	12	16	22	50	79	114	8
7	JFK	FLL	6:00	8:55	23	23	25	29	33	34	39	78	8

Continued on next page

Table B.1 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
8	BOS	IAD	17:35	19:20	0	0	0	10	23	39	52	148	8
11	ORD	JFK	11:15	14:55	0	0	0	2	23	48	61	118	8
26	JFK	IAD	14:25	15:50	15	17	18	18	27	28	73	88	8
28	JFK	HOU	19:00	22:24	0	0	4	26	29	56	116	122	8
38	BOS	RDU	13:20	15:32	0	0	0	0	9	24	27	116	8
41	PIT	JFK	5:50	7:25	0	0	0	0	0	16	42	138	8
44	RIC	JFK	9:25	10:37	0	0	0	0	10	17	33	103	8
45	TPA	BOS	18:30	21:31	6	7	25	27	31	45	68	80	8
51	JFK	ACK	16:35	18:13	23	27	28	37	47	50	66	122	8
55	JFK	CMH	15:15	17:23	0	0	0	7	18	41	62	134	8
62	JFK	PWM	18:55	20:41	10	14	18	19	25	30	66	96	8
63	JFK	BNA	20:15	22:05	0	26	32	33	36	66	73	77	8
65	JFK	PIT	20:05	22:07	2	3	25	31	35	39	55	75	8
68	IAD	BOS	10:10	11:43	0	0	0	6	9	55	64	112	8
74	JFK	PIT	11:10	12:48	5	6	6	24	24	52	92	100	8
75	JFK	BOS	13:10	14:28	0	0	3	38	50	56	66	148	8
80	IAD	JFK	7:25	8:40	0	7	8	17	22	25	39	129	8
84	JFK	ORD	8:45	10:30	17	19	25	29	49	63	87	97	8
85	JFK	RDU	9:05	10:47	25	25	30	32	43	52	70	129	8
86	JFK	RIC	7:10	8:47	0	0	0	28	36	47	62	135	8
88	JFK	RIC	16:10	18:05	9	10	16	38	54	55	62	92	8
89	HOU	JFK	6:00	10:26	0	0	0	15	17	17	54	86	8
90	JFK	RDU	16:45	18:57	20	20	23	26	31	35	37	104	8
94	MCO	IAD	20:20	22:30	0	0	2	4	8	11	56	108	8
102	AUS	JFK	6:00	10:40	0	0	0	19	21	21	35	83	8
103	AUS	BOS	11:55	17:00	0	0	12	16	19	39	60	88	8
104	FLL	HPN	9:35	12:30	6	15	17	17	21	22	34	126	8
39	PIT	JFK	9:30	11:00	0	0	0	0	0	103	212	429	9
81	IAD	JFK	11:45	12:57	0	2	12	13	39	181	368	385	9
13	CLT	JFK	16:15	18:15	54	58	74	85	133	156	192	394	10
17	CMH	JFK	17:55	19:47	43	64	67	71	76	123	131	281	10
35	PIT	JFK	18:25	20:02	52	69	69	83	95	98	135	257	10
47	RIC	JFK	13:30	14:48	18	23	43	57	77	118	143	345	10

Table B.2: Cluster Label for Each flight in N_2

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
4	OAK	LGB	11:25	12:47	0	0	0	0	0	0	0	0	1
5	BOS	JFK	6:50	8:05	0	9	16	17	28	29	71	386	1

Continued on next page

Table B.2 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
10	ORD	JFK	8:15	11:55	0	0	0	0	0	0	0	0	1
22	PBI	JFK	9:35	12:14	0	0	0	0	0	10	31	278	1
23	BOS	MCO	5:30	8:30	0	0	0	0	0	0	0	7	1
31	SYR	JFK	5:00	6:12	0	0	0	0	0	15	67	426	1
33	LAS	LGB	18:05	19:12	0	0	0	0	0	3	4	36	1
36	SMF	LGB	9:50	11:12	0	0	0	0	0	0	0	37	1
40	LAS	LGB	13:10	14:18	0	0	0	0	0	0	3	17	1
43	SMF	LGB	16:00	17:23	0	0	0	0	0	0	0	7	1
45	LAS	LGB	6:00	7:05	0	0	0	0	0	0	0	0	1
53	PWM	JFK	5:00	6:25	0	0	0	0	0	0	5	33	1
59	ROC	JFK	7:30	8:45	0	0	0	0	0	0	1	24	1
71	MCO	BOS	9:25	12:18	0	0	0	0	0	0	0	22	1
75	LGB	LAS	12:20	13:30	0	0	0	0	0	0	0	15	1
82	MCO	EWR	7:00	9:34	0	0	0	0	0	0	0	40	1
92	LGB	SMF	7:50	9:10	0	0	0	0	0	0	0	8	1
96	BOS	DEN	20:35	22:57	0	0	0	0	0	0	2	28	1
2	EWR	MCO	17:35	20:37	0	0	0	0	10	11	23	28	2
12	BOS	MCO	13:00	16:05	17	22	32	32	49	57	93	120	2
16	JFK	SYR	14:20	15:38	7	11	15	15	34	85	193	270	2
19	PBI	HPN	16:15	19:05	18	18	19	20	24	25	32	34	2
29	SWF	MCO	13:50	16:50	3	10	12	27	37	47	86	93	2
50	TPA	JFK	5:05	7:45	0	0	0	0	2	6	22	23	2
51	TPA	JFK	10:40	13:17	0	0	0	0	7	20	42	59	2
52	PWM	JFK	15:10	16:25	94	138	139	149	153	183	281	282	2
57	SYR	JFK	16:15	17:25	36	37	45	57	64	98	222	237	2
61	BQN	MCO	3:00	5:46	0	1	2	2	3	3	8	10	2
62	ROC	JFK	16:10	17:29	24	25	33	57	112	115	230	242	2
69	JFK	BOS	14:30	16:13	0	13	19	36	53	90	162	260	2
72	MCO	BOS	16:45	19:40	63	73	75	75	80	106	142	145	2
81	MCO	EWR	13:35	16:36	104	104	153	175	186	186	279	326	2
84	JFK	BUF	19:50	21:39	25	28	28	29	45	56	91	113	2
90	LGB	SLC	10:35	13:15	3	4	4	4	6	10	17	21	2
99	FLL	JFK	12:35	15:24	15	19	27	28	104	121	314	424	2
9	JFK	FLL	20:45	0:02	9	10	15	17	20	38	48	56	3
25	BUF	JFK	16:00	17:25	15	38	41	100	122	129	189	209	3
26	BOS	SEA	17:10	20:19	2	15	21	26	46	61	62	67	3
35	JFK	BUR	17:20	20:46	0	0	0	28	37	45	47	50	3
42	LGA	FLL	8:30	11:35	0	0	0	0	17	28	35	35	3

Continued on next page

Table B.2 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
44	LAS	JFK	14:25	22:34	0	0	33	53	74	76	87	131	3
48	JFK	BUF	13:55	15:23	0	0	0	3	15	22	23	30	3
55	JFK	DEN	18:55	21:35	0	0	9	17	36	50	60	79	3
58	LGA	MCO	20:00	22:51	0	0	0	0	2	2	3	3	3
65	JFK	PSE	22:50	2:48	7	11	12	19	25	26	27	44	3
67	LGB	OAK	15:00	16:16	0	0	3	3	4	6	9	13	3
76	LGB	LAS	16:05	17:15	0	0	0	1	3	4	6	8	3
77	LGB	LAS	18:05	19:15	0	1	4	5	9	14	15	18	3
100	MCO	JFK	13:00	15:37	50	72	80	81	97	146	154	178	3
103	MCO	JFK	6:40	9:09	0	0	0	7	24	31	32	59	3
106	MCO	JFK	17:30	20:10	96	98	128	135	169	170	176	177	3
107	AUS	JFK	16:20	21:15	33	35	59	100	145	145	152	152	3
108	JFK	SFO	20:10	23:35	32	43	52	58	59	67	70	76	3
109	FLL	EWR	11:55	15:03	43	90	107	195	206	243	247	362	3
14	BOS	OAK	7:00	10:30	0	0	0	0	0	0	13	19	4
20	JFK	JAX	12:40	15:10	2	12	17	31	33	39	145	179	4
28	SWF	MCO	10:50	13:45	0	0	0	0	0	1	15	32	4
38	RSW	JFK	9:35	12:17	0	0	0	0	2	6	91	121	4
73	JFK	ORD	5:45	7:30	0	0	0	0	0	0	34	42	4
78	JFK	BQN	4:45	8:23	0	0	0	0	0	0	27	47	4
79	JFK	BTV	21:55	23:18	3	16	32	34	35	42	155	183	4
86	HPN	MCO	19:55	22:40	0	0	0	0	0	0	77	80	4
88	JFK	PBI	12:25	15:18	0	0	5	11	13	26	90	127	4
89	HPN	MCO	6:55	9:40	0	0	0	0	0	0	53	72	4
97	FLL	LGA	5:05	7:50	0	0	0	0	0	0	15	32	4
105	MCO	PSE	22:10	1:05	0	0	0	0	0	0	3	3	4
115	MCO	SWF	10:20	13:10	0	0	0	0	0	1	18	26	4
1	EWR	MCO	10:15	12:54	0	0	0	1	22	30	33	90	5
11	JFK	LAS	9:45	12:10	0	21	23	25	34	54	70	183	5
18	JFK	TPA	7:00	10:00	0	0	0	0	7	24	38	82	5
21	JFK	JAX	7:05	9:38	11	16	17	22	47	61	100	171	5
27	BUF	JFK	19:10	20:35	19	25	26	46	52	59	74	133	5
39	SLC	LGB	14:05	14:55	0	0	0	0	3	4	5	14	5
41	RSW	JFK	16:50	19:45	39	45	54	73	88	125	158	245	5
47	JFK	BUF	9:50	11:14	12	17	32	35	35	38	47	75	5
54	JFK	AUS	12:35	15:38	0	0	0	0	0	6	25	58	5
56	JAX	JFK	15:50	18:10	32	64	70	72	96	104	126	259	5
60	ROC	JFK	18:45	20:05	28	30	38	53	62	62	72	120	5

Continued on next page

Table B.2 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
66	JFK	PWM	13:10	14:30	2	4	16	21	25	63	73	207	5
68	JFK	PBI	6:00	8:51	0	0	0	16	18	24	30	71	5
74	MCO	BQN	23:25	2:03	6	6	8	11	11	15	32	56	5
83	JFK	BTV	8:15	9:35	17	24	28	31	31	46	74	126	5
85	MCO	HPN	15:25	18:05	32	32	34	42	44	73	135	236	5
87	HPN	PBI	18:55	21:40	0	0	1	3	3	11	24	45	5
93	LGB	SMF	14:00	15:20	0	0	0	0	0	6	8	24	5
94	JFK	RSW	13:00	16:08	6	11	17	18	31	32	59	104	5
98	JFK	ROC	14:05	15:30	5	8	15	29	35	65	98	215	5
101	JFK	ROC	8:25	9:50	19	27	27	34	40	61	63	144	5
102	MCO	JFK	5:05	7:35	0	0	0	0	1	16	19	36	5
104	MCO	LGA	16:30	19:18	42	48	53	67	83	112	120	184	5
113	MSY	JFK	9:50	13:40	0	0	0	0	0	9	12	30	5
114	JFK	MCO	9:30	12:10	28	34	42	47	49	57	59	95	5
116	JFK	MCO	20:35	23:45	24	24	25	35	39	42	56	108	5
117	MCO	SWF	7:30	10:10	0	0	0	0	2	8	8	19	5
3	EWR	PBI	15:55	18:52	3	3	5	5	14	21	26	127	6
6	JFK	FLL	18:05	21:30	23	30	44	46	50	53	67	163	6
7	OAK	LGB	17:00	18:20	0	0	0	0	0	0	3	13	6
8	JFK	FLL	8:45	11:42	36	36	37	47	56	111	116	461	6
13	ORD	JFK	14:35	17:50	36	39	58	83	84	97	105	445	6
15	JFK	TPA	18:15	21:31	11	15	25	28	31	67	78	288	6
17	JFK	SYR	8:25	9:43	34	38	40	42	46	52	84	239	6
24	BUF	JFK	11:50	13:05	1	3	4	11	22	38	47	177	6
30	JAX	JFK	10:20	12:29	0	2	8	16	36	58	92	361	6
32	SYR	JFK	10:30	11:35	0	3	3	8	11	13	14	62	6
34	IAD	LGB	7:20	9:45	0	0	0	0	0	15	24	121	6
37	PBI	JFK	19:55	22:37	0	0	0	0	0	3	16	66	6
46	JFK	BUF	16:20	18:12	0	15	15	32	42	52	53	282	6
49	BTV	JFK	10:25	11:40	0	0	3	3	5	8	12	37	6
63	BQN	JFK	9:20	13:00	0	0	0	0	8	10	30	91	6
64	ROC	JFK	10:40	11:52	0	0	1	1	3	15	28	82	6
70	JFK	ORD	12:15	13:53	11	12	13	20	36	41	56	279	6
80	JFK	ROC	16:10	17:58	3	19	21	31	36	53	68	264	6
91	LGB	SLC	19:00	21:45	0	0	0	0	2	4	15	74	6
95	JFK	RSW	5:50	8:55	2	8	10	16	19	20	91	418	6
110	JFK	MSY	6:55	9:06	0	4	5	15	18	19	52	177	6
111	JFK	MCO	18:10	21:18	43	43	56	60	66	69	81	203	6

Continued on next page

Table B.2 – continued from previous page

Flight	Ori	Des	Local time-SD	Local time-SA	Eight largest primary delays								Custer label
112	JFK	MCO	13:00	15:47	35	38	49	58	73	77	150	466	6

BIBLIOGRAPHY

Adan, I. J., Kulkarni, V. G., Lee, N., and Lefeber, E. (2018). Optimal routeing in two-queue polling systems. *Journal of Applied Probability*, 55(3):944–967.

Adusumilli, K. M. and Hasenbein, J. J. (2010). Dynamic admission and service rate control of a queue. *Queueing Systems*, 66(2):131–154.

Ageeva, Y. (2000). *Approaches to incorporating robustness into airline scheduling*. PhD thesis, Massachusetts Institute of Technology.

Ahmadbeygi, S., Cohn, A., and Lapp, M. (2010). Decreasing airline delay propagation by re-allocating scheduled slack. *IIE transactions*, 42(7):478–489.

Airlines for America (2019). Passenger carrier delay costs. https://www.airlines.org/dataset/per-minute-cost-of-delays-to-u-s-airlines.

Aldous, D. (1992). Asymptotics in the random assignment problem. *Probability Theory and Related Fields*, 93(4):507–534.

Antunes, D., Vaze, V., and Antunes, A. P. (2019). A robust pairing model for airline crew scheduling. *Transportation Science*.

Arıkan, M., Deshpande, V., and Sohoni, M. (2013). Building reliable air-travel infrastructure using empirical data and stochastic models of airline networks. *Operations Research*, 61(1):45–64.

Armony, M., Plambeck, E., and Seshadri, S. (2009). Sensitivity of optimal capacity to customer impatience in an unobservable m/m/s queue (why you shouldn't shout at the dmv). *Manufacturing & Service Operations Management*, 11(1):19–32.

Baidari, I. and Sajjan, S. (2016). International journal of mathematical archive-7 (10), 2016, 123-127 available online through www. ijma. info issn 2229–5046.

Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., Zou, B., et al. (2010). Total delay impact study. In *NEXTOR Research Symposium, Washington DC*.

Barnhart, C. and Cohn, A. (2004). Airline schedule planning: Accomplishments and opportunities. *Manufacturing & service operations management*, 6(1):3–22.

Barnhart, C., Fearing, D., and Vaze, V. (2014). Modeling passenger travel and delays in the national air transportation system. *Operations Research*, 62(3):580–601.

Bekker, R. and Borst, S. C. (2006). Optimal admission control in queues with workload-dependent service rates. *Probability in the Engineering and Informational Sciences*, 20(4):543–570.

Borgs, C., Chayes, J. T., Doroudi, S., Harchol-Balter, M., and Xu, K. (2014). The optimal admission threshold in observable queues with state dependent pricing. *Probability in the Engineering and Informational Sciences*, 28(1):101.

Borndörfer, R., Dovica, I., Nowak, I., and Schickinger, T. (2010). Robust tail assignment.

Borst, S., Mandelbaum, A., and Reiman, M. I. (2004). Dimensioning large call centers. *Operations research*, 52(1):17–34.

Burkard, R. E., Klinz, B., and Rudolf, R. (1996). Perspectives of monge properties in optimization. *Discrete Applied Mathematics*, 70(2):95–161.

Cachon, G. P. and Zipkin, P. H. (1999). Competitive and cooperative inventory policies in a two-stage supply chain. *Management science*, 45(7):936–953.

Chen, H. and Solak, S. (2015). Lower cost arrivals for airlines: Optimal policies for managing runway operations under optimized profile descent. *Production and Operations Management*, 24(3):402–420.

Cho, S.-H. and Tang, C. S. (2013). Advance selling in a supply chain under uncertain supply and demand. *Manufacturing & Service Operations Management*, 15(2):305–319.

Chowdhury, S., Schulz, E., Milner, M., and Van De Voort, D. (2014). Core employee based human capital and revenue productivity in small firms: An empirical investigation. *Journal of Business Research*, 67(11):2473–2479.

Chr, N. (1972). Individual and social optimization in a multiserver queue with a general cost-benefit structure. *Econometrica: Journal of the Econometric Society*, pages 515–528.

Clarke, L., Johnson, E., Nemhauser, G., and Zhu, Z. (1997). The aircraft rotation problem. *Annals of Operations Research*, 69:33–46.

Deshpande, V. and Arıkan, M. (2012). The impact of airline flight schedules on flight delays. *Manufacturing & Service Operations Management*, 14(3):423–440.

Dong, L. and Rudi, N. (2004). Who benefits from transshipment? exogenous vs. endogenous wholesale prices. *Management Science*, 50(5):645–657.

Dunbar, M., Froyland, G., and Wu, C.-L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Science*, 46(2):204–216.

Dunbar, M., Froyland, G., and Wu, C.-L. (2014). An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers & Operations Research*, 45:68–86.

D'Auria, B. and Kanta, S. (2015). Pure threshold strategies for a two-node tandem network under partial information. *Operations Research Letters*, 43(5):467–470.

Edelson, N. M. and Hilderbrand, D. K. (1975). Congestion tolls for poisson queuing processes. *Econometrica: Journal of the Econometric Society*, pages 81–92.

Ehrgott, M. and Ryan, D. M. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150.

Eltoukhy, A. E., Chan, F. T., and Chung, S. H. (2017). Airline schedule planning: A review and future directions. *Industrial Management & Data Systems*.

Emami, P., Pardalos, P. M., Elefteriadou, L., and Ranka, S. (2018). Machine learning methods for solving assignment problems in multi-target tracking. *arXiv preprint arXiv:1802.06897*.

Estes, A. S. and Ball, M. O. (2021). Monge properties, optimal greedy policies, and policy improvement for the dynamic stochastic transportation problem. *INFORMS Journal on Computing*, 33(2):785–807.

Froyland, G., Maher, S. J., and Wu, C.-L. (2013). The recoverable robust tail assignment problem. *Transportation Science*, 48(3):351–372.

Gao, C., Johnson, E., and Smith, B. (2009). Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1):2–16.

Guo, P. and Zipkin, P. (2007). Analysis and comparison of queues with different levels of delay information. *Management Science*, 53(6):962–970.

Guthrie, J. P. (2001). High-involvement work practices, turnover, and productivity: Evidence from new zealand. *Academy of management Journal*, 44(1):180–190.

Halfin, S. and Whitt, W. (1981). Heavy-traffic limits for queues with many exponential servers. *Operations research*, 29(3):567–588.

Hassin, R. and Haviv, M. (2003). *To queue or not to queue: Equilibrium behavior in queueing systems*, volume 59. Springer Science & Business Media.

Haviv, M. (2014). Regulating an m/g/1 queue when customers know their demand. *Performance Evaluation*, 77:57–71.

Haviv, M. and Oz, B. (2016). Regulating an observable m/m/1 queue. *Operations Research Letters*, 44(2):196–198.

Haviv, M. and Oz, B. (2018). Self-regulation of an unobservable queue. *Management Science*, 64(5):2380–2389.

Hu, X. and Ralph, D. (2007). Using epecs to model bilevel games in restructured electricity markets with locational prices. *Operations research*, 55(5):809–827.

Huselid, M. A. (1995). The impact of human resource management practices on turnover, productivity, and corporate financial performance. *Academy of management journal*, 38(3):635–672.

Janssen, A. and van Leeuwaarden, J. S. (2015). Staffing many-server systems with admission control and retrials. *Advances in Applied Probability*, 47(2):450–475.

Kang, L. S. (2004). *Degradable airline scheduling: an approach to improve operational robustness and differentiate service quality*. PhD thesis, Massachusetts Institute of Technology.

Koçağa, Y. L. and Ward, A. R. (2010). Admission control for a multi-server queue with abandonment. *Queueing Systems*, 65(3):275–323.

Köchel, P. (2004). Finite queueing systems—structural investigations and optimal design. *International Journal of Production Economics*, 88(2):157–171.

Koole, G. and Pot, A. (2011). A note on profit maximization and monotonicity for inbound call centers. *Operations research*, 59(5):1304–1308.

Kouvelis, P. and Zhao, W. (2012). Financing the newsvendor: supplier vs. bank, and the structure of optimal trade credit contracts. *Operations research*, 60(3):566–580.

Krokhmal, P. A. and Pardalos, P. M. (2009). Random assignment problems. *European Journal of Operational Research*, 194(1):1–17.

Kulkarni, V. G. (2016). *Modeling and analysis of stochastic systems*. Crc Press.

Laboratory Corporation of America (2020). Specimen collection and shipping instructions. https://www.avancecare.com/wp-content/uploads/2020/03/Labcorp-COVID-19-NP-OP-Specimen-Collection-and-Shipping-Instructions.pdf.

Lan, S., Clarke, J.-P., and Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation science*, 40(1):15–28.

Lee, H. L. and Cohen, M. A. (1983). A note on the convexity of performance measures of m/m/c queueing systems. *Journal of Applied Probability*, pages 920–923.

Lee, J., Marla, L., and Jacquillat, A. (2020). Dynamic disruption management in airline networks under airport operating uncertainty. *Transportation Science*, 54(4):973–997.

Liang, L., Wu, J., Cook, W. D., and Zhu, J. (2008). The dea game cross-efficiency model and its nash equilibrium. *Operations research*, 56(5):1278–1288.

Liu, L. and Kulkarni, V. G. (2006). Explicit solutions for the steady state distributions in m/ph/1 queues with workload dependent balking. *Queueing Systems*, 52(4):251–260.

Liu, L. and Kulkarni, V. G. (2008a). Balking and reneging in m/g/s systems exact analysis and approximations. *Probability in the Engineering and Informational Sciences*, 22(3):355.

Liu, L. and Kulkarni, V. G. (2008b). Busy period analysis for m/ph/1 queues with workload dependent balking. *Queueing Systems*, 59(1):37–51.

Liu, S. W., Thomas, S. H., Gordon, J. A., Hamedani, A. G., and Weissman, J. S. (2009). A pilot study examining undesirable events among emergency department–boarded patients awaiting inpatient beds. *Annals of emergency medicine*, 54(3):381–385.

Mandelbaum, A. and Zeltyn, S. (2009). Staffing many-server queues with impatient customers: constraint satisfaction in call centers. *Operations research*, 57(5):1189–1205.

Marla, L., Vaze, V., and Barnhart, C. (2018). Robust optimization: Lessons learned from aircraft routing. *Computers & Operations Research*, 98:165–184.

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

Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476.

Messerli, E. (1972). Bstj brief: Proof of a convexity property of the erlang b formula. *The Bell System Technical Journal*, 51(4):951–953.

Naor, P. (1969). The regulation of queue size by levying tolls. *Econometrica: journal of the Econometric Society*, pages 15–24.

Nash Jr, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49.

Official Airline Guide (2020). Punctuality league 2020 report. https://www.oag.com/punctuality-league-2020-report.

Patel, P. B., Combs, M. A., and Vinson, D. R. (2014). Reduction of admit wait times: the effect of a leadership-based program. *Academic Emergency Medicine*, 21(3):266–273.

Pines, J. M., Iyer, S., Disbot, M., Hollander, J. E., Shofer, F. S., and Datner, E. M. (2008). The effect of emergency department crowding on patient satisfaction for admitted patients. *Academic Emergency Medicine*, 15(9):825–831.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Ramdas, K. and Williams, J. (2006). An empirical investigation into the tradeoffs that impact on-time performance in the airline industry. *Washington Post*, pages 1–32.

Rosenberger, J. M., Johnson, E. L., and Nemhauser, G. L. (2004). A robust fleet-assignment model with hub isolation and short cycles. *Transportation science*, 38(3):357–368.

Sanders, J., Borst, S., Janssen, A., and Leeuwaarden, J. v. (2017). Optimal admission control for many-server systems with qed-driven revenues. *Stochastic Systems*, 7(2):315–341.

Schumer, C. and Maloney, C. B. (2008). Your flight has been delayed again: flight delays cost passengers, airlines, and the us economy billions. *The US Senate Joint Economic Committee*.

Shi, P., Chou, M. C., Dai, J. G., Ding, D., and Sim, J. (2016). Models and insights for hospital inpatient operations: time-dependent ED boarding time. *Management Science*, 62(1):1–28.

Shortle, J. F., Thompson, J. M., Gross, D., and Harris, C. M. (2018). *Fundamentals of queueing theory*, volume 399. John Wiley & Sons.

Singer, A. J., Thode Jr, H. C., Viccellio, P., and Pines, J. M. (2011). The association between length of emergency department boarding and mortality. *Academic Emergency Medicine*, 18(12):1324–1329.

Smith, B. C. and Johnson, E. L. (2006). Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516.

Smith, J. M. (2003). M/g/c/k blocking probability models and system performance. *Performance Evaluation*, 52(4):237–267.

Smith, J. M., Cruz, F., and van Woensel, T. (2010). Optimal server allocation in general, finite, multi-server queueing networks. *Applied Stochastic Models in Business and Industry*, 26(6):705–736.

Stidham Jr, S. (2009). *Optimal design of queueing systems*. CRC press.

Stidham Jr, S. and Weber, R. R. (1989). Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations research*, 37(4):611–625.

Ward, A. R. and Kumar, S. (2008). Asymptotically optimal admission control of a queue with impatient customers. *Mathematics of Operations Research*, 33(1):167–202.

Wei, K. and Vaze, V. (2018). Modeling crew itineraries and delays in the national air transportation system. *Transportation Science*, 52(5):1276–1296.

Weide, O., Ryan, D., and Ehrgott, M. (2010). An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 37(5):833–844.

Yadav, R., Khanna, A., Panday, P., Dasmohapatra, S., et al. (2019). An analytical study of quality of work life & organisational commitment and their relation with revenue per employee of major it companies in india. *Journal of Human Resource and Sustainability Studies*, 7(02):284.

Yan, C. and Kung, J. (2016). Robust aircraft routing. *Transportation Science*, 52(1):118–133.

Yechiali, U. (1971). On optimal balking rules and toll charges in the gi/m/1 queuing process. *Operations Research*, 19(2):349–370.

Yen, J. W. and Birge, J. R. (2006). A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14.