OPTIMAL ADMISSION CONTROL IN TANDEM AND PARALLEL QUEUEING SYSTEMS WITH APPLICATIONS TO COMPUTER NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Daniel F. Silva

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2016

OPTIMAL ADMISSION CONTROL IN TANDEM AND PARALLEL QUEUEING SYSTEMS WITH APPLICATIONS TO COMPUTER NETWORKS

Approved by:

Dr. Hayriye Ayhan, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Sigrún Andradóttir
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Antonius Dieker
Department of Industrial Engineering and Operations Research
Columbia University

Dr. David Goldberg
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Bo Zhang
T. J. Watson Research Center
IBM Research

Date Approved: 7/22/2016

To my parents,

Carlos F. Silva & María Inés Izquierdo.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to all the people who made this dissertation, and my Ph.D. journey, possible. First and foremost, I would like to thank my advisor, Prof. Hayriye Ayhan, for her guidance and encouragement throughout my studies. I am extremely grateful for Hayriye's brilliant insights, her dedication and her seemingly infinite patience, without which, this thesis would never have been finished. I am also deeply thankful to Dr. Bo Zhang, who has practically served as a co-advisor for this dissertation. Without Bo's ideas, thoroughness and keen eye for detail this work would not be what it is today.

This dissertation, as well as my experience at Georgia Tech, have also been greatly enriched by the remaining members of my committee: Profs. David Goldberg, Ton Dieker and Sigrún Andradóttir. David and Ton were both exceptional teachers, all I know about the foundations of Stochastic processes, I learned from them. They have also, throughout long talks about the future, become mentors for me, especially helping me navigate the next steps in my academic career, for this I am very grateful. I am also very thankful to Sigrún for her encouragement and her insightful comments on my research, which have helped improve it considerably.

Of course, this research would not have been possible without the ISYE faculty. I would especially like to thank Profs. Tovey, Nemhauser, Monteiro, Kim, Kleywegt, Ahmed and Dai, as well as Profs. Popescu, Lounici and Iliev at the Department of Mathematics, for the wonderful classes I had the opportunity to take from them. I would also like to thank Profs. Gary Parker, Paul Kvam, Alan Erera and Edwin Romeijn for giving me the continued opportunity and support to succeed in the graduate program, here at ISYE. Thank you, also, to all the amazing ISYE staff, for making my life so much easier.

I would be remiss, if I failed to thank my professors at Universidad de Los Andes for laying the groundwork to make this journey possible. I am especially grateful to my friend and mentor, Dr. Germán Riaño, who encouraged me to pursue a doctoral degree in the first place.

I would also like to thank my friends and colleagues, who made the past five years a fun and enjoyable experience. I will forever cherish the memories we made with my Atlanta family, grilling at the house on 16th street, traveling around the U.S. and the world, as well as discovering this lovely city together; they are Diego Morán, Tim Sprock, Margaret Johnston, Andrés Iroume, Stefanía Stefánsdóttir, Mallory Solder, Matthias Klapp, Francisca Otero and Vinod Cheriyan. Their constant support and unconditional friendship kept me sane. A huge thanks also to all my fiends here at ISYE and beyond, including Alfredo, Cristobal, Gustavo, Tamara, Alvaro, Alejandra, Rodolfo, Pamela, Camilo, Tatiana, Kevin, Lauren, Jeff, Tonya, Ethan, Toyya, Brian, Matt, James, Ben, Erin, Carl, Ezgi, Ilke, Murat, Bahar, Isil, Tugce, Norbert, Jan, Min-Kyoung, Chang-Han, and all the others, which are too many to list here.

Finally, a very special thank you to my family, especially my parents, Carlos and María Inés. Without their perpetual support and encouragement nothing I have done would ever have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Modern computer networks require advanced, efficient algorithms to control several aspects of their operations, including routing data packets, access to secure systems and data, capacity and resource allocation, task scheduling, etc. A particular class of problems that arises frequently in computer networks is that of admission and routing control. Two areas where admission control problems are common are traffic control and authentication procedures. This thesis focuses on developing tools to solve problems in these areas. We begin the thesis with a brief introductory chapter describing the problems we will be addressing. Then, we follow this with a review of the relevant literature on the problems we study and the methodologies we use. Then, we have the main body of the dissertation, which is divided into three parts, described below.

In the first part, we analyze a problem related to data routing in a network. Specifically, we study the problem of admission control to a system of two stations in tandem with finite buffers, Poisson arrivals to the first station, and exponentially distributed service times at both stations. We assume costs are incurred either when a customer is rejected at the time of arrival to the first station or when the second station is full at the time of service completion at the first station. We propose a Markov decision process formulation for this problem. Then, we use this model to show that, when one of the buffers has size one, the structure of the optimal policy is threshold and that only two particular policies can be optimal. We provide the exact optimality thresholds for small systems. For larger systems, we formulate heuristic policies and use numerical experiments to show that these policies achieve

near-optimal performance.

For the second part of this thesis, we investigate the system described above in a more general case, where the capacity of the buffers at both station is equal, finite and arbitrary. We focus on two specific, extremal policies, which we call the Prudent and Greedy policies. We derive a closed-form expression for the long-run average reward under the Prudent policy and provide a necessary and sufficient threshold condition for it to be optimal. For the Greedy policy, we give a matrix-analytic solution for the long-run average reward and provide a sufficient condition for it to be optimal. We also prove that it is always optimal to admit customers in the states where the Prudent policy admits customers. Next, we use an example to illustrate that the optimal policy can have a complicated form. Finally, we propose two heuristic policies and use numerical experiments to show that they perform much better than the Prudent and Greedy policies, and in fact, achieve near-optimal performance.

In the third and final part of this dissertation, we shift our attention to a different admission and routing control problem. We study a centralized system where requests for authentication arrive from different users. The system has multiple authentication methods available and a controller must decide how to assign a method to each request. We consider three different performance measures: usability, operating cost, and security. First, we model the problem using a cost-based approach, which assigns a cost to each measure of performance. Under this approach, we find that if each authentication method has infinitely many servers the optimal policy is static and deterministic. On the other hand, if there is one method that has finite capacity and the rest have infinitely many servers, we show that the optimal policy is of trunk reservation form. Then, we model the problem using a constraint-based approach, which assumes hard constraints on some of the measures of performance. We show that if each method has infinitely many servers, the optimal policy is static and randomized. While, if one method has finite capacity and the rest have infinitely many servers, we show that the optimal policy has a 2-randomized trunk reservation form. Finally, we illustrate how to use our results to construct an efficient frontier of non-dominated solutions.

We end this dissertation with a short recapitulation of our main contributions and a discussion on potential avenues for future research.

CHAPTER I

INTRODUCTION

In recent years computer networks have become ubiquitous in daily life. A local computer network such as Georgia Tech's consists of thousands of interconnected computers, which have to handle traffic from thousands of users simultaneously, transmitting millions of packets of data every day. The Internet as whole is much larger and more complex. Just to get a sense of its scope, consider what happens in the Internet globally in a single minute, illustrated in Figure 1.

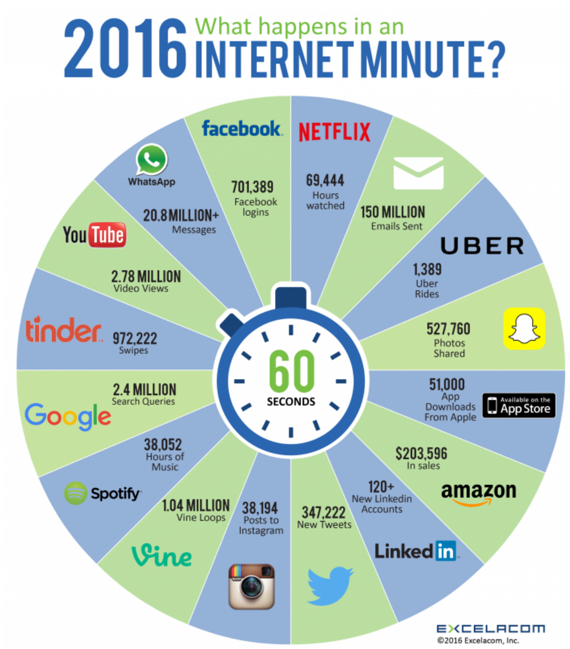

Figure 1: What happens in an internet minute? LaBouef [41]

These massive and complex networks require advanced, efficient algorithms to control several aspects of their operations, including routing data packets, access to secure systems and data, capacity and resource allocation, task scheduling, etc.

There are several characteristics of computer networks that make them amenable to stochastic modeling, and in particular to queueing models. First, there is an inherent variability in the volume, timing, and duration of the operations that are required. Also, data over these networks is handled in packets, and requests and tasks occur one at a time, therefore the flow of customers in these systems should be modeled discretely. Finally, the packets or tasks are routed through complex networks, where at each station they must wait their turn to be processed. Considering this combination of characteristics, queueing networks have been one of the preferred tools of researchers to model complex computer networks, as they capture all of these aspects.

A particular class of problems that arises frequently in computer networks and naturally lends itself to a queueing theoretic approach is that of admission and routing control. Such problems can appear in many different operations related to managing these networks. Two areas where admission control problems are common are traffic control and authentication procedures. In certain situations, the traffic control problem consists of deciding whether to admit new packets into a sequence of servers. This problem can be modeled as admission control into a tandem queueing network. Similarly, a typical step in authentication processes consists of choosing the right authentication server (from several available servers), which can be modeled as routing in a network of parallel queueing systems. This thesis focuses on developing tools to solve these problems. Next, we describe each of them in more detail.

A common problem in traffic control is to decide whether to admit or reject incoming data packets, or customers in a particular branch or section of a network, which is controlled locally. The administrator of the local network wishes to choose a policy for deciding whether or not to admit an incoming packet with the objective of minimizing the system's overall operating cost. We focus on a special class of networks, called tandem loss networks, which are common in telecommunications (see Bertsekas and Gallagher [5] and the references therein). These networks consist of several finite-capacity stations in series, where a packet which is admitted must be processed sequentially at each station. In a tandem loss network blocking is not allowed, but rather, if a customer completes service at a given station and the subsequent station is full, then the customer is ejected from the system and a penalty cost (which depends on the station) is incurred by the system administrator. The administrator can also choose to reject an incoming customer and incur a (different) penalty cost. In this thesis we study this problem analytically in order to gain an understanding of the optimal admission control policies.

Although tandem loss networks arise naturally in telecommunications, recently, similar systems have been used to model multiple situations in other applications, such as call centers (see for example Li and Whitt [46]), and healthcare (see for example Litvak, van Rijsbergen, Boucherie and van Houdenhoven [49]). Therefore, an analytical solution could be applied in practice to those other application areas as well. Currently, there is little work found in the literature on admission control in tandem loss networks.

In the first part of this dissertation, we analyze the problem of admission control to a tandem loss network with two stations. Specifically, we study a system of two queues in tandem with finite buffers, Poisson arrivals to the first station, and exponentially distributed service times at both stations. We assume a rejection cost is incurred each time a customer is rejected at the first station and a loss cost is incurred each time a customer is ejected from the system because the second station is full at the time of service completion at the first station. We model the system for arbitrary finite buffers using queueing theory and Markov Decision Processes (MDP). In order to better understand optimal policies, we start with the case where one of the stations has no waiting space, that is, only one customer can be at the station at a time. This problem in itself presents several challenges. For this reason, we will dedicate an entire chapter to it. We show that when there is a unitary buffer at one of the stations, the structure of the optimal policy is threshold and that only two particular policies can be optimal. We provide the exact optimality thresholds for small systems. For larger systems, we formulate heuristic policies and use numerical experiments to show that these policies achieve near-optimal performance.

Next, in the second part of this dissertation, we investigate the more general case, where the size of the buffer at both stations is equal, finite and arbitrary. We focus on two specific, extremal policies, for which we provide a way to efficiently calculate the long-run average reward, as well as necessary and/or sufficient optimality conditions. We also provide analysis on the behavior of these two policies as the buffer sizes go to infinity. Then, we prove a structural result, which all optimal policies must meet. We also offer a discussion on the characteristics of the optimal policy as the size of the buffers increases. Finally, we propose two heuristic policies and show, through numerical experiments, that these achieve near-optimal performance.

Then, we shift our attention to a different admission and routing control problem. Many online services require some form of user authentication to grant access to secure data. Most authentication systems have multiple authentication schemes available, including various forms of biometric data acquisition, behavioral analyses, and schemes that consider contextual factors (see Bao, Pierce, Whittaker and Zhai [4]). These different schemes have very different resource requirements, wait times and also differ in the level of security that each affords. The system administrator would like to minimize the perceived inconvenience to the user, as well as maximizing the authentication confidence, all while keeping operating costs low.

This type of system can be modeled as an admission and routing control problem in a queueing network consisting of several parallel queueing systems. However, in this particular application there are multiple objectives. While there is some literature regarding admission and routing in parallel systems, there is little work for the case with multiple objectives. There is also, to the best of our knowledge, no previous work on modeling an authentication system using queueing theory.

For the third and final portion of this dissertation, we study a centralized system where requests for authentication arrive from different users. The system has multiple authentication methods available and a controller must decide how to assign a method to each request. We model each authentication method as a multi-server queue and model the trade-offs between three different performance measures: operating cost, latency and security. We use two different approaches: first, a cost based approach, that gives different weights to each objective, and then a constraint-based approach, which assumes hard restrictions on some of the performance measures. For each approach we construct a mathematical model and derive structural and computational results on the optimal admission and routing policy. We also provide a numerical example to illustrate how a system administrator can use the models we propose to build an efficient frontier of non-dominated solutions and select an appropriate one.

The rest of this document is organized as follows: Chapter 2 provides a review of the relevant literature. We focus on work related to the methodologies applied throughout this thesis, in several application areas. In Chapter 3, we study the problem of admission control in a tandem loss system with two stations. We model the problem as a MDP and characterize the optimal policy when either of the stations has a buffer of size one. In Chapter 4, we consider a more general version of the tandem loss system with two stations, where the the buffers at both stations are of finite and equal size. We analyze the performance of two particular policies and give optimality conditions for those policies. We also discuss the structure of the optimal policy and present numerical results for some heuristic policies. In Chapter 5, we present the authentication method assignment problem. We model it mathematically and provide two approaches to find policies that balance the three performance measures. Finally, in Chapter 6, we summarize the main contributions of this thesis and present some extensions and open questions that we plan to explore in the future. Appendix A provides some supplemental material for Chapter 3.

CHAPTER II

LITERATURE REVIEW

Most of the literature on queueing networks, where each station has finite capacity, focuses on blocking paradigms (see for example Perros [56], and Balsamo, de Nitto Personé, and Onvural [3]). However, in this thesis we focus on two classes of networks where blocking does not occur. These are tandem loss networks and parallel queueing networks. In particular, we deal with admission and routing in these kinds of networks. In this chapter, we present the relevant literature related to performance evaluation and control of tandem queueing networks (with and without losses), and parallel queueing networks.

Evaluating performance of purely tandem networks has gotten special attention in the literature, including analytical results, such as Grassmann and Drekic [23], asymptotic results, as in Martin [50] or approximation methods, as in Perros and Altiok [57], and the references therein. The more specific case of networks consisting of several tandem lines in parallel have been studied either as routing problems (see for example Gosavi and Smith [22]) or as capacity allocation problems (see for example Daskalaki and Smith [14]). However, none of these authors consider the loss feature described in Chapter 1.

Loss networks have been studied in telecommunications literature. However, most of the work has focused on a particular problem where an incoming customer simultaneously uses multiple resources over a network. As an example, consider connecting a phone call through multiple links, where several links are required for the call to go through. If the call is connected, the customer uses all the necessary links simultaneously and releases them all upon service completion. In these systems, losses are said to occur when a call cannot be connected for lack of capacity in some links. Note that, in such a telephone network, customers do not go from station to station, but rather use several resources in the system at once. For a comprehensive survey of these types of loss networks see Kelly [32]. Control problems in this context have been addressed using techniques from queueing theory and Markov decision processes such as Hunt and Laws [29], Kelly, Key and Zachary [33] and Key [34], just to name a few. However, the loss networks described in Chapter 1 are different from these, because customers travel the system one station at a time and losses occur at any point in the network. Thus, the results in this literature are not applicable to the problem we are considering.

Most of the literature dealing with admission control to queueing networks focuses on networks with infinite capacity buffers or blocking. However, several related control problems in tandem queues can be found. Stidham and Weber [69] give a good survey of applications of Markov decision processes to admission control problems in queueing networks.

Ku and Jordan [38] study admission control policies for a system of two multi-server Markovian queues with loss. However, the system they consider has no waiting space, the capacity is the same as the number of servers. They allow two types of customers: type 1 customers are more valuable and require service at both stations with a positive probability; and type 2 customers require service only at the second station. They show that to maximize the total discounted reward over an infinite horizon, type 1 customers must always be accepted and there is a threshold for admission of type 2 customers. Later, Ku and Jordan [40] generalize these results for the case with n customer types and provide admission control heuristics that achieve near optimal performance. Also, Ku and Jordan [39] develop admission control policies for a multi-server loss queue which is itself fed by an upstream set of parallel multi-server loss queues and by a stream of customers from outside the system. Here, the objective is to choose the number of servers to reserve for each customer stream. Revenue is gained by each customer and at the target queue revenue depends on the source of the customer. They prove that the policy that maximizes total discounted revenue consists of a set of monotonically decreasing thresholds (as functions of the occupancy of each queue). They obtain a fluid limit as the number of lines goes to infinity and solve the related optimization problem. Chang and Chen [9] also consider a system with no waiting space, that is, the capacity is equal to the number of servers, specifically, a two stage tandem system where any customer who finds all servers busy at its destination queue is lost, and compare the loss rates under several admission control policies.

A different approach is taken by Ghoneim and Stidham [20], who consider admission control to a system of two tandem queues with infinite buffers. Their objective is to maximize the discounted expected net benefit over a finite or infinite horizon, where net benefit is composed of random rewards for entering customers minus holding costs at each queue. Hordijk and Koole [28] consider a similar system, where each station has two servers, each with an independent queue. They find a policy which stochastically minimizes the number in the system at any given time for certain system parameters and show that choosing the shortest queue may not be optimal. Also, Hordijk and Koole [27], consider routing to parallel queues in which each queue has its own single server and service times are exponential with non-identical parameters. Here the queues may have finite buffers, and the arrival process can be controlled and can depend on the state and routing policy. They give conditions on the cost function such that the optimal policy always assigns customers to the faster queue. Zhang and Philles [76] consider two infinite capacity queues in tandem, each of which has its own input of arriving customers, which in turn, may either be accepted or rejected. They suppose that the system receives a fixed reward for each accepted customer and pays a holding cost per customer per unit time in the system. They use fuzzy control to determine long-run average reward optimal policies.

Leskelä and Resing [43] analyze a two station tandem queue with infinite capacity and a control policy determined by whether the queue at the second station exceeds a fixed threshold. They give conditions for stability, analyze the steady state distribution of the queue lengths using matrix-analytic methods and solve numerically for the average sojourn time and throughput. They conclude that these measures are non-monotone with respect to the service rates. Finally, Kim and Dudin [36] expand on Leskelä and Resing [43]. They analyze a two-stage multi-server tandem queue with two types of customers, where priority customers are always admitted to the system. Non-priority customers are admitted to the system only if the number of busy servers at the second stage does not exceed some pre-assigned threshold. They find the stationary distribution and calculate performance measures with respect to the threshold, numerically.

Spicer and Ziedins [68] are the first to consider admission control for the type of loss networks described in Chapter 1. They consider a system of several tandem loss lines in parallel. In their work, each line is made up of two single server finite capacity queues. They show that it may be optimal for an arriving customer to select a queue with more customers in order to minimize its individual loss probability. Sheu and Ziedins [63] consider admission and routing control in a system of N parallel tandem lines where each line is a set of two single server queues with finite buffers and losses can occur. They obtain the fluid limit as $N \to \infty$ and solve the related optimization problem. They find that the asymptotically optimal policy takes one of two forms, either accept arrivals into any line where there is capacity in the first station, or only accept arrivals if there is a line where they cannot be lost. The assymptotically optimal policy is threshold type and depends on the ratio between the cost of rejecting an incoming arrival and the loss cost after service at the first station. Ziedins [77] studies the differences between the optimal policies from the two previous papers and gives numerical examples showing that for some small systems the user-optimal and system-optimal policies may differ and that as the service rate is increased at the second stage the user-optimal policy may change in such a way that the total expected cost due to loss increases.

To the best of our knowledge the first paper to provide exact analytical results for an optimal long-run average cost policy for tandem queues with loss was Zhang and Ayhan [75]. They consider a system with a finite arbitrary buffer at the first station and a unitary buffer at the second station. They showed that the optimal policy is of threshold type. In Chapters 3 and 4, we build on the results of Zhang and Ayhan [75], providing several results for the case with a unitary buffer at the first station and a finite buffer at the second station, as well as a more general case with finite buffers of equal size at both stations.

Now, let us consider the relevant literature for assigning costumers in a network consisting of several servers in parallel. There are several classic results related to this problem. The seminal work of Winston [73] considers a system of several identical, exponential single-server queues in parallel and identical customers that arrive following a Poisson process, and shows that a join-the-shortest-queue (JSQ) policy minimizes both the customer's expected delay and the average delay in the system. Weber [71] extends this result for any arrival process and service time distributions with a non-decreasing hazard rate. Whitt [72] shows that if the service time distribution does not have non-decreasing hazard rate, then a JSQ policy may not minimize the customer's expected delay, and also, that a strategy that minimizes a customer's delay may not minimize the average delay in the system. Johri [30] shows that a JSQ policy minimizes each customer's expected delay in the case where the service times are exponential and have non-decreasing state-dependent service rates. Hordijk and Koole [26] prove that a JSQ policy is also optimal for the case of finite buffers and batch arrivals.

The previous results all focus on the case where customers are identical, no jockeying is allowed between queues and decisions are made deterministically. Several variations on this problem have been studied, specially in the context of load-balancing or server assignment in parallel computer systems. For a survey on queueing methods applied to load balancing, routing and server assignment in parallel and distributed systems see Boxma, Koole and Liu [8]. For a more recent and broader view (not limited to queueing theoretic methods) see Baccelli, Jean-Marie, and Mitrani [2].

A case of particular interest to our research is that of multiple customer classes and several parallel queues. This has also been studied in the literature, specially in load balancing applications. Ni and Hwang [53] consider Poisson arrivals from m customer classes that must be assigned to n parallel $M/M/1$ queues and propose an assignment policy that minimizes the average wait time. Bonomi and Kumar [7] consider a variation of that problem where each queue has a dedicated arrival stream, and allow the service time distributions to be general. They develop heuristic algorithms and derive the optimal policy when the customers are homogeneous. Sethuraman and Squillante [62] consider a similar problem. However, they consider each station as a $M/G/1$ queue and the objective in this case is to minimize a linear function of the per-class average response time. Furthermore, they consider a class of policies that performs static routing to each queue and sequencing within the queue. Their objective is to find optimal policies within this class. Ansell, Glazebrook and Kirkbride [1] consider the same problem, but allow dynamic routing and scheduling processes. They introduce a family of policies that can be constructed algorithmically and achieve near-optimal performance.

Most of the work mentioned so far focuses on reducing waiting times or queue lengths. Our research, on the other hand, focuses on admission and routing control of multiple customer classes to parallel systems, where each customer class has an associated reward and/or costs. The classic reference in this case is Miller [51], which considers a single $M/M/c/c$ queue with several customer classes, where each class has a reward associated for service completion. Miller [51] concludes that a trunk-reservation policy ordered by the rewards maximizes the long-run average reward. Lewis, Ayhan and Foley [44] show that for a single, finite-capacity queueing system, with Poisson arrivals, monotone state-dependent exponential service rates and multiple customer classes with rewards, a trunk reservation policy is again optimal. Lewis, Ayhan and Foley [45] extend those results to include non-stationary rates and capacities. Lin and Ross [47] provide conditions for a trunk reservation policy to be optimal even in the case the gatekeeper has incomplete information. Feinberg and Reiman [19] show that for a similar system (with full information), but with the addition of a constraint on the blocking probability of the most profitable customer, a trunk reservation policy where one of the thresholds is randomized is optimal. We should note that this problem was initially proposed in Reiman [59], where its solution was conjectured. Fan-Orzechowski and Feinberg [17] generalize this result by adding penalty costs charged for each rejection and having a single constraint on the average penalty cost per unit time. They reach the same conclusion as Feinberg and Reiman [19]. Fan-Orzechowski and Feinberg [18] generalize the previous result to allow an arbitrary k number of constraints. In this case, the optimal policy is shown to be a trunk reservation policy where k of the thresholds are randomized. In Chapter 5, we build on several of these results to define the structure of optimal policies for the authentication method assignment problem, treating it alternatively as a multi-objective problem with weights for each objective, and as a single-objective problem with constraints.

CHAPTER III

ADMISSION CONTROL FOR A TANDEM LOSS SYSTEM WITH TWO STATIONS

In this chapter, we study the problem of admission control into a tandem system of two stations as a Markov Decision Process (MDP). We consider a system with two stations in tandem and one server at each station. Arrivals to the system follow a Poisson Process with rate λ and service times at each station follow an exponential distribution with rate μ_i, $i = 1, 2$. Each station has a finite capacity denoted by $B_i < \infty$, $i = 1, 2$. Upon each arrival a gatekeeper has to decide (based on full knowledge of the state of the system) whether to admit or reject the incoming arrival. If an arrival is not admitted, a cost c_1 is incurred. If a customer completes service at the first station and at that time the second station is full, the customer is lost and a penalty cost of c_2 is incurred. Note that if the first station is full at the time of an arrival, then the incoming customer has to be rejected and a cost of c_1 is incurred. Our objective is to determine an admission control policy at the first station that minimizes the long-run average cost. In this chapter, we provide a MDP formulation for the system with general $1 \leq B_i < \infty$, $i = 1, 2$. Then we use this model to derive analytical results for the specific cases $B_1 = 1, B_2 < \infty$ and $B_1 < \infty, B_2 = 1$. Then in Chapter 4 we consider a case with identical, arbitrary, finite buffers at both stations. Figure 2 illustrates the system we are studying for the particular case of $B_1 = 3, B_2 = 4$.

The remainder of this chapter is organized as follows: in Section 3.1 we begin by providing a MDP formulation of the tandem loss system described above. Section 3.2 presents the first of two special cases, namely we consider the system under the

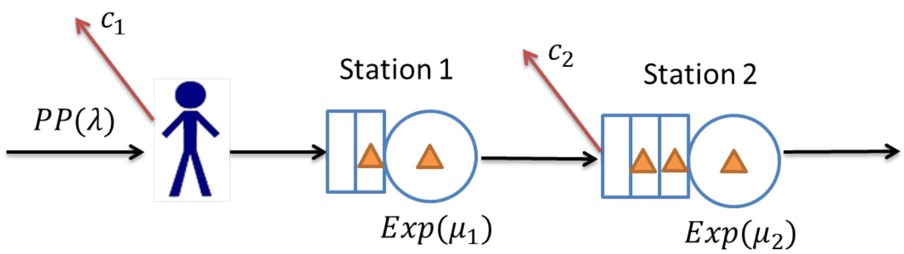

Figure 2: A tandem loss system with two stations.

assumption $B_2 = 1$. This section summarizes previous findings of Zhang and Ayhan [75]. Afterwards, in Section 3.3 we focus on the specific case where $B_1 = 1$ and we show that there are only two policies that could be optimal depending on a threshold on the ratio $\frac{c_2}{c_1}$. Next, we provide closed form expressions for the threshold when $B_2 \leq 10$. However, since the characterization of a closed form expression for general values of B_2 is difficult, we provide heuristics that serve as an alternative to the threshold policy. We also present numerical results that show that the heuristics have near-optimal performance in most cases. Finally, Section 3.4 presents our conclusion. This chapter is based on Silva, Zhang and Ayhan [65]. Supplemental material for this chapter is presented in Appendix A.

3.1 Markov Decision Process Formulation for General Buffer Sizes

In this section, using uniformization (see Lippman [48]), we formulate the admission control problem as a discrete-time MDP. Furthermore, we show that we have a *unichain* model. Suppose that the two servers work at all times. The service at a station when there is no customer is referred to as *fictitious* service. Then, we let the gatekeeper make an admission decision *right before* the occurrence of every event. There are three types of events: a (real or fictitious) service completion at station 1, a service completion at station 2, or a customer arrival. If the event that occurs right after a decision is made is an arrival, the decision applies; otherwise, the decision does not have any effect on the system. So, the times between consecutive decision epochs are independent exponential random variables with rate $\mu_1 + \mu_2 + \lambda$. Minimizing the continuous-time long-run average cost of this system is equivalent to minimizing the long-run average cost of the discrete-time MDP over this new set of decision epochs.

Without loss of generality, we assume $\mu_1 + \mu_2 + \lambda = 1$ in the following analysis. Let $\mathbf{1}_{\{X\}} = 1$ if condition X holds, and $\mathbf{1}_{\{X\}} = 0$ otherwise. We then define the following discrete-time Markov decision process problem.

We have a discrete-time Markov chain with state space $S = \{(q_1, q_2) \in \mathbb{Z}^2 : 0 \leq q_1 \leq B_1, 0 \leq q_2 \leq B_2\}$, where q_i denotes the number of customers at station i (including those waiting and in service), $i = 1, 2$. The size of the state space is $|S| = (B_1 + 1) \cdot (B_2 + 1)$.

Let 0 denote rejecting the next potential arrival and 1 denote accepting the next potential arrival. Then, the sets of allowable actions are $A_{(q_1,q_2)} = \{0\}$ if $q_1 = B_1$, and $A_{(q_1,q_2)} = \{0, 1\}$, for any other $(q_1, q_2) \in S$. Let $p(\cdot|(q_1, q_2), d)$ denote the transition probability when action d is taken in state (q_1, q_2). We have

$$p(\boldsymbol{s}|(q_1, q_2), d) = \begin{cases} \mu_1, \text{ if } \boldsymbol{s} = ((q_1 - 1)^+, q_2 + \mathbf{1}_{\{q_1 > 0 \text{ and } q_2 < B_2\}}) \\ \mu_2, \text{ if } \boldsymbol{s} = (q_1, (q_2 - 1)^+) \\ \lambda, \quad \text{if } \boldsymbol{s} = (q_1 + \mathbf{1}_{\{d=1\}}, q_2). \end{cases} \tag{1}$$

with $p((0, 0)|(0, 0), 0) = 1$. Let $r(\boldsymbol{s}, d)$ denote the expected reward received when action d is taken in state $\boldsymbol{s}$. Specifically,

$$r((q_1, q_2), d) = -c_1\lambda\mathbf{1}_{\{d=0\}} - c_2\mu_1\mathbf{1}_{\{q_1>0 \text{ and } q_2=B_2\}}. \tag{2}$$

Let $\mathbf{r}_\pi$ be the vector of rewards for each state under policy π. Similarly define P_π as the transition probability matrix of the discrete-time Markov chain (DTMC) resulting from implementing policy π. It is known that any policy π^* that solves the Optimality Equations for the MDP is also a long-run average reward optimal policy.

The optimality equations for this MDP are:

$$0 = \max_{\pi \in \Pi} \{\mathbf{r}_\pi + ge + (P_\pi - I)h\} \tag{3}$$

where g is a scalar and represents the long-run average reward, h is the bias vector (as defined on page 338 of Puterman [58]) and e is a vector of ones.

We will refer to the above mathematical model as the MDP model throughout this chapter and Chapter 4. As mentioned before, our goal is to find a policy π^*, which maximizes the long-run average reward g. We wish to understand the structure of such a policy, and how it changes with respect to the state and system parameters.

As introduced in Puterman [58], a discrete-time MDP is *unichain*, if the resulting Markov chain under every deterministic stationary policy has a single recurrent class plus a possibly empty set of transient states. In fact, the discrete-time MDP described above for our model is unichain. This can be seen from the following argument. Regardless of the policy and the initial state, state $(0, 0)$ can be reached with probability 1, due to a long enough interarrival time. Therefore, $(0, 0)$ is recurrent, and all the states accessible from $(0, 0)$, together with $(0, 0)$, form a recurrent class; other states form a set of transient states.

The existence of a deterministic stationary optimal policy for our model is guaranteed by the finiteness of the state space and the action space, together with the fact that the model is unichain and the rewards are bounded (see Theorem 8.4.5 in Puterman [58]). Therefore, for each state we need only consider deterministic decision rules. We assume the convention that, a policy π is binary-valued function $\pi : S \to \{0, 1\}$. So from here on we only refer to policies and not decision rules, this is a common convention (see Guo and Hernandez-Lerma [24]). We refer to the optimal policy as $\pi^*$and to the optimal long-run average reward as $g^*(B_1, B_2)$.

Let $S_d := \{(q_1, q_2) \in S : 0 \leq q_1 \leq B_1 - 1\}$ be the set of states at which a choice of acceptance or rejection needs to be made. Each possible combination of actions to be taken at each state in S_d comprises a deterministic policy π.

3.1.1 The Prudent and Greedy Policies

In this section, we introduce two policies that are intuitive and easy to implement. Later, we show that if one of the buffers is unitary, then these are the only two policies which can be optimal. The first policy is one that only admits an arrival if the probability of loss at the second station is zero. The second policy admits an arrival as long as there is space in the first buffer.

Define a *Prudent policy* (π_P) as a policy that only admits an arrival whenever there are fewer customers in the system than the capacity of the buffer at the second station. This policy never incurs the loss cost. Furthermore, of the policies that never incur the loss cost, this policy admits incoming customers most often. That is, for state $s = (q_1, q_2)$ we have

$$\pi_P(s) = \begin{cases} 1 & q_1 + q_2 < B, \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, consider a policy which accepts every arrival if possible (that is, if there is room in the buffer at the first station). We call this a *Greedy policy* (π_G). So, for state $s = (q_1, q_2)$ we have

$$\pi_G(s) = \begin{cases} 1 & q_1 < B, \\ 0 & \text{otherwise.} \end{cases}$$

The next two sections consider the special cases $B_1 < \infty, B_2 = 1$ and $B_1 = 1, B_2 < \infty$ respectively. In those sections we show that the Prudent policy and the Greedy policy are the only two policies that can be optimal and provide necessary and sufficient conditions for each of them to be optimal. We also provide expressions for the long-run average gain as a function of the buffer sizes. We define $g_P(B_1, B_2)$ as the gain under the Prudent policy and $g_G(B_1, B_2)$ as the gain under the Greedy policy.

3.2 Case 1: When B_2=1

Note that in this case the Prudent policy reduces to admitting a new arrival if the system is empty upon arrival, and rejecting in all other cases. And the Greedy policy admits in all $s \in S_d$. This case was studied in detail by Zhang and Ayhan [75]. We provide their main result here for completeness (but omit the proof).

Theorem 1 (Zhang and Ayhan)**.** *Define*

$$c_*(1) = \left(1 + \frac{\mu_2^2}{\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2}\right),$$

then

1. *If $\frac{c_2}{c_1} \geq c_*(1)$, the Prudent policy is optimal, and it is the unique stationary optimal policy when the inequality is strict. Furthermore, in this case*

$$g^*(B_1, 1) = g_P(B_1, 1) = c_1 \frac{\lambda^2(\mu_1 + \mu_2)}{\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2}.$$

2. *If $\frac{c_2}{c_1} \leq c_*(1)$, the Greedy policy is optimal, and it is the unique stationary optimal policy when the inequality is strict. Furthermore, in this case*

$$g^*(B_1, 1) = g_G(B_1, 1) =$$
$$c_1 \frac{\lambda^{B_1+1}(\mu_1 - \lambda)}{\mu_1^{B_1+1} - \lambda^{B_1+1}} + c_2 \frac{\lambda^2\mu_1^2\left[\left(\mu_1^{B_1} - \lambda^{B_1}\right) + \mu_2\left(\mu_1^{B_1-1} - \lambda^{B_1-1}\right)\right]}{(\lambda + \mu_2)(\mu_1 + \mu_2)\left(\mu_1^{B_1+1} - \lambda^{B_1+1}\right)}.$$

3.3 Case 2: When B_1=1

Here we focus on the original problem with $B_1 = 1$ and $B_2 < \infty$. An example of a system under these conditions is illustrated in Figure 3.

Observing the structure of S_d for this case, one can deduce that a deterministic policy that rejects at $(0, q_2)$ for some $q_2 < B_2$ cannot be optimal, as any arrival admitted into the system at such states will not be lost. So, the system would incur a cost of c_1 when there is zero probability of incurring c_2. In the next proposition, we formalize this idea.

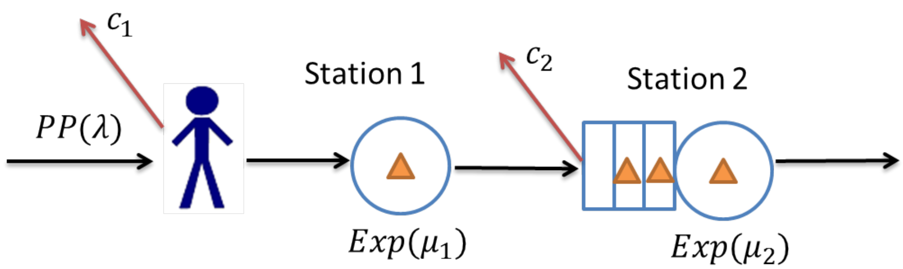

Figure 3: Tandem loss system with two stations where $B_1 = 1, B_2 = 4$

Proposition 1. *For a given policy π it is always optimal to accept at $(0, q_2)$ for any $q_2 < B_2$. Or conversely a deterministic policy that rejects at $(0, q_2)$ for some $q_2 < B_2$ cannot be optimal.*

Proof. We prove by induction. For $q_2 = 0$, it is true for the following reason: rejecting at (0,0) yields the long-run average cost λc_1 per time unit; accepting at (0,0) while rejecting at all other states yields $\lambda c_1 \delta$, where δ, the long-run fraction of time that the resulting system is not in state (0,0), must be strictly less than 1.

Now suppose the desired result holds for $q_2 = 0, 1, ..., j-1$, where $j < B_2$. Then we show it is also true for $q_2 = j$.

Consider two systems: system I under any policy π, under which the rejection action is taken at state $(0, j)$ and system II also under π except that the acceptance action is taken at state $(0, j)$. Both systems start with the same initial state.

Because, for system I, all states (q_1, q_2) with $q_1 + q_2 \geq j + 1$ are transient, we can assume that the policy π prescribes the rejection action for all these states. Also, by the induction hypothesis, it suffices for us to assume that the policy π prescribes the acceptance action for $(0, q_2)$, where $q_2 = 0, 1, ..., j-1$. Note that both assumptions are effective for system II as well, because it also operates under π.

Consider any sample path. If state $(0, j)$ is never seen by an arrival on this sample path, both systems evolve identically. We now show that, if $(0, j)$ is ever seen by an arrival at some point in time, system II will have no more cost than system I from that point, say T_1, to the next time point when both systems reach the same state, say T_2.

First, we note that no loss ever occurs at station 2 in either system because $j < B_2$ and there are never more than $j+1$ customers in either system. Second, at any point in time between T_1 and T_2 the states of the two systems at a customer arrival epoch must have one of the following forms.

- System I at $(0, j)$ and system II at $(1, j)$. The arrival is rejected in both systems in this case.

- System I at $(0, i)$ and system II at $(0, i+1)$, where $i < j$. In this case, the arrival is accepted in both systems by the induction hypothesis and also, for $i = j-1$, using the assumption that the acceptance action is taken at state $(0, j)$ in system II.

- System I at $(0, j)$ and system II at $(0, j+1)$. In this case, the arrival is rejected in both systems.

- System I at $(0, i)$ and system II at $(1, i)$, for some $i < j$. The arrival is accepted in system I by induction hypothesis but rejected in system II. Between T_1 and T_2 this can only occur once, after which both systems immediately reach the same state.

Therefore, from just after T_1 to T_2, at most c_1 more cost is incurred in system II than in system I. Also, because, at time T_1, system I has a rejection cost c_1 while system II does not have any cost, we conclude that system II has no more cost than system I from T_1 to T_2. Because π is arbitrary, it is optimal to accept at $(0, j)$. ■

Proposition 1 implies that for a given value of B_2 and a known set of parameters either the Prudent policy must be optimal or the Greedy policy must be optimal. This leaves the question of determining which of the two policies is optimal. In order to answer this question, we compute the long-run average reward under both policies and compare the gain values. Recall $g_P(1, B_2)$ and $g_G(1, B_2)$ are the long-run average gain of operating a system with buffer B_2 under π_P and π_G, respectively. Comparing $g_P(B_1, B_2)$ and $g_P(B_1, B_2)$ we can immediately conclude that there exists a threshold $c_*(B_2)$ such that if $\frac{c_2}{c_1} > c_*(B_2)$ then π_P is optimal and otherwise π_G is optimal.

Let $p_{(q_1,q_2)}$, $(q_1, q_2) \in S$ be the stationary distribution of the continuous-time Markov chain (CTMC) model of this system operating under the Prudent policy, that is $S = \{(q_1, q_2) \in \mathbb{Z}^2 : 0 \le q_1 \le 1, 0 \le q_2 \le B_2\}$. Define the transition rates $\gamma(i, j)$ for $i, j \in S$ as

$$\gamma(i,j) = \begin{cases} \lambda, \text{ for } i = (0, q_2),\ j = (1, q_2),\ 0 \le q_2 < B_2, \\ \mu_1, \text{ for } i = (1, q_2),\ j = (0, q_2 + 1),\ 0 \le q_2 < B_2, \\ \mu_2, \text{ for } i = (q_1, q_2),\ j = (q_1, q_2 - 1),\ q_1 \in \{0, 1\}\ 1 \le q_2 \le B_2, \\ 0 \text{ otherwise }. \end{cases}$$

Similarly, let $\hat{p}_{(q_1,q_2)}$, $(q_1, q_2) \in S$ be the stationary distribution of the resulting CTMC when operating under the Greedy policy π_G, that is, a model with the same state space and transition rates as above, except $\gamma((0, B_2), (1, B_2)) = \lambda$ instead of zero. Define:

$$c_*(B_2) = \frac{\lambda}{\mu_1} \left(\frac{\sum_{i=0}^{B_2-1} \left(p_{(1,i)} - \hat{p}_{(1,i)}\right) + p_{(0,B_2)}}{\hat{p}_{(1,B_2)}} - 1 \right).$$

Theorem 2. *If, for a fixed buffer size B_2, $\frac{c_2}{c_1} > c_*(B_2)$ then $g_P(1, B_2) > g_G(1, B_2)$ and π_P is optimal for this system, otherwise π_G is optimal.*

Proof. Note, as a result of the Prudent policy $p_{(0,B_2)} = 0$. Then, for fixed B_2:

$$g_P(1, B_2) = -c_1\lambda \left(\sum_{i=0}^{B_2-1} p_{(1,i)} + p_{(0,B_2)} \right).$$

Likewise for the Greedy policy:

$$g_G(1,B_2) = -c_1\lambda\sum_{i=0}^{B_2}\hat{p}_{(1,i)} + c_2\mu_1\hat{p}_{(1,B_2)}.$$

The result then immediately follows from Proposition 1 and the comparison of $g_P(1,B_2)$ and $g_G(1,B_2)$. ■

In the next section we provide closed form expressions for $c_*(B_2)$ when $B_2 \leq 10$.

3.3.1 Characterization of $c_*(B_2)$

First consider the case where $B_2 = 1$. Note that this is a special case of the model developed by Zhang and Ayhan [75]. Our derivation and that in Zhang and Ayhan [75] both yield

$$c_*(1) = \left(1 + \frac{\mu_2^2}{\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2}\right).$$

For $B_2 = 2$, the gain for π_P, is

$$g_P(1,2) = -c_1\frac{\lambda^2\left(\mu_2^2\left(\mu_1+\mu_2\right) + \lambda\left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right)\right)}{\mu_1\mu_2^2(\mu_1+\mu_2) + \lambda\mu_2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2+\mu_1\mu_2+\mu_2^2\right)}.$$

Similarly, the gain for π_G, is

$$g_G(1,2) = \\ -\frac{\lambda^2\left(c_2\lambda\mu_1^3 + c_1\left(\mu_2^2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2+\mu_1\mu_2+\mu_2^2\right) + \lambda\mu_2\left(\mu_1^2 + 2\mu_1\mu_2^2 + 2\mu_2^2\right)\right)\right)}{(\lambda+\mu_1)\left(\mu_2^2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2+\mu_1\mu_2+\mu_2^2\right) + \lambda\mu_2\left(\mu_1^2+2\mu_1\mu_2+2\mu_2^2\right)\right)}.$$

Hence for $B_2 = 2$,

$$c_*(2) = \left(1 + \frac{\mu_2^3\left(\lambda+\mu_1+\mu_2\right)}{\mu_1\mu_2^2(\mu_1+\mu_2) + \lambda\mu_2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2+\mu_1\mu_2+\mu_2^2\right)}\right).$$

For larger values of $B_2 \leq 10$, following the procedure described above we calculate the threshold cost $c_*(B_2)$, obtaining expressions of the form

$$c_*(B_2) = \left(1 + \frac{\alpha(B_2)}{\beta(B_2)}\right),$$

where for $B_2 \leq 10$,

$$\alpha(B_2) = \mu_2^{B_2+1} \sum_{i=1}^{\lceil B_2/2 \rceil} \Big[\big(\lambda^{B_2-i} + \mathbf{1}_{\{(B_2-i)>(i-1)\}} \lambda^{i-1} (\mu_1+\mu_2)^{B_2-2i+1} \big)$$
$$\sum_{j=0}^{i-1} \left(\mu_1^{i-j-1} \mu_2^{j} \binom{i-1}{j} \frac{1}{j!} \prod_{k=1}^{j} (B_2 - i + k) \right) \Big].$$

Closed form expressions for $\beta(B_2)$ for $B_2 = 1, ..., 10$ can be found in Appendix A, in Section A.1. After the publication of Silva *et al.* [65] a closed form expression for $p_{(q_1,q_2)}$ and $\hat{p}_{(q_1,q_2)}$ for a general B_2 was provided in Kim and Kim [35].

In the next two sections we take a heuristic approach to understand the optimal policy and the optimal gain as $B_2 \to \infty$.

3.3.2 Trends in probabilities

We want to understand the impact of choosing the wrong policy on the optimal cost. To do this, we investigate the behavior of the probabilities of being in states where the the Prudent policy and the Greedy policy are different. These next three propositions follow by computing the stationary distributions of the corresponding continuous time Markov chain.

Proposition 2. *Under π_P the stationary probability of being in state $(0, B_2)$, $p_{(0,B_2)}$ is strictly decreasing in B_2 for $0 \leq B_2 \leq 10$.*

Proof. The stationary probabilities for each $B_2 \leq 10$ can be written as

$$p_{(0,B_2)} = \frac{(\lambda\mu_1)^{B_2}}{(\lambda\mu_1)^{B_2-1}(\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2) + f(\lambda, \mu_1, \mu_2, B_2)},$$

where $f_1(\lambda, \mu_1, \mu_2, B_2)$ is a polynomial function of the parameters which depends on the buffer size. More specifically,

$$
\begin{aligned}
f_1(\lambda, \mu_1, \mu_2, 1) &= 0, \\
f_1(\lambda, \mu_1, \mu_2, 2) &= \lambda^2\mu_2^2 + 2\lambda\mu_1\mu_2^2 + \mu_1^2\mu_2^2 + \lambda\mu_2^3 + \mu_1\mu_2^3, \\
f_1(\lambda, \mu_1, \mu_2, 3) &= \lambda^3\mu_1\mu_2^2 + 2\lambda^2\mu_1^2\mu_2^2 + \lambda\mu_1^3\mu_2^2 + \lambda^3\mu_2^3 + 3\lambda^2\mu_1\mu_2^3 + 3\lambda\mu_1^2\mu_2^3 + \mu_1^3\mu_2^3 \\
&\quad + 2\lambda^2\mu_2^4 + 4\lambda\mu_1\mu_2^4 + 2\mu_1^2\mu_2^4 + \lambda\mu_2^5 + \mu_1\mu_2^5.
\end{aligned}
$$

The remaining values of $f_1(\lambda, \mu_1, \mu_2, B_2)$ are provided in the Appendix, in Section A.2. Note that the expression

$$\lambda\mu_1 f_1(\lambda, \mu_1, \mu_2, B_2) < f_1(\lambda, \mu_1, \mu_2, B_2 + 1)$$

holds for the values of B_2 presented above. The same inequality holds for any $B_2 \leq 10$. We can verify this with the expressions for $f_1(\lambda, \mu_1, \mu_2, B_2)$ in Section A.2. This implies that $p(0, B_2)$ is strictly decreasing in B_2 for π_P for any fixed set of parameters.

■

Now consider the Greedy policy.

Proposition 3. *Under π_G the stationary probability of being in state $(0, B_2)$, $\hat{p}_{(0,B_2)}$ is strictly decreasing in B_2 for $0 \leq B_2 \leq 10$.*

Proof. The stationary probabilities for each $B_2 \leq 10$ can be expressed as

$$\hat{p}_{(0,B_2)} = \frac{(\lambda\mu_1)^{B_2}(\mu_1 + \mu_2)}{(\lambda\mu_1)^{B_2-1}(\mu_1 + \mu_2)\left(\lambda^2 + \lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2\right) + f_2(\lambda, \mu_1, \mu_2, B_2)},$$

where $f_2(\lambda, \mu_1, \mu_2, B_2)$ is a polynomial function of the parameters which depends on the buffer size. More specifically,

$$
\begin{aligned}
f_2(\lambda, \mu_1, \mu_2, 1) &= 0, \\
f_2(\lambda, \mu_1, \mu_2, 2) &= \lambda^3\mu_2^2 + 2\lambda^2\mu_1\mu_2^2 + 2\lambda\mu_1^2\mu_2^2 + \mu_1^3\mu_2^2 + 2\lambda^2\mu_2^3 + 4\lambda\mu_1\mu_2^3 + 2\mu_1^2\mu_2^3 \\
&\quad + \lambda\mu_2^4 + \mu_1\mu_2^4,
\end{aligned}
$$

$$
\begin{aligned}
f_2(\lambda,\mu_1,\mu_2,3) = {} & \lambda^4\mu_1\mu_2^2 + 2\lambda^3\mu_1^2\mu_2^2 + 2\lambda^2\mu_1^3\mu_2^2 + \lambda\mu_1^4\mu_2^2 + \lambda^4\mu_2^3 + 4\lambda^3\mu_1\mu_2^3 + 6\lambda^2\mu_1^2\mu_2^3 \\
& + 4\lambda\mu_1^3\mu_2^3 + \mu_1^4\mu_2^3 + 3\lambda^3\mu_2^4 + 8\lambda^2\mu_1\mu_2^4 + 8\lambda\mu_1^2\mu_2^4 + 3\mu_1^3\mu_2^4 + 3\lambda^2\mu_2^5 \\
& + 6\lambda\mu_1\mu_2^5 + 3\mu_1^2\mu_2^5 + \lambda\mu_2^6 + \mu_1\mu_2^6.
\end{aligned}
$$

The remaining values of $f_2(\lambda,\mu_1,\mu_2,B_2)$ are provided in Appendix A, in Section A.3. We observe that for any fixed set of parameters the expression

$$
\lambda\mu_1 f_2(\lambda,\mu_1,\mu_2,B_2) < f_2(\lambda,\mu_1,\mu_2,B_2+1)
$$

holds for the values of B_2 presented above. The same inequality holds for any $B_2 \leq 10$. We can verify this with the expressions for $f_2(\lambda,\mu_1,\mu_2,B_2)$ in Section A.3. This implies that $\hat{p}_{(0,B_2)}$ is strictly decreasing in B_2 for π_G. ■

Finally, consider state $(1,B_2)$ under the Greedy policy, where cost c_2 may be incurred.

Proposition 4. *Under π_G the stationary probability of being in state $(1,B_2)$, $\hat{p}_{(1,B_2)}$ is strictly decreasing in B_2 for $0 \leq B_2 \leq 10$.*

Proof. For $B_2 \leq 10$ we have,

$$
\hat{p}_{(1,B_2)} = \frac{\lambda^{B_2+1}\mu_1^{B_2}}{(\lambda\mu_1)^{B_2-1}(\mu_1+\mu_2)\left(\lambda^2+\lambda\mu_1+\lambda\mu_2+\mu_1\mu_2\right)+f_2(\lambda,\mu_1,\mu_2,B_2)}.
$$

Note that the denominator of this expression is the same as the one for $\hat{p}_{(0,B_2)}$ so the result holds. ■

The above propositions lead us to the following conjecture:

Conjecture 1. *Under π_P as $B_2 \to \infty$, $p_{(0,B_2)}$ converges monotonically to some value $p^*_{(0,B_2)} \geq 0$. Under π_G as $B_2 \to \infty$ the probabilities $\hat{p}_{(0,B_2)}$ and $\hat{p}_{(1,B_2)}$ converge monotonically to values $\hat{p}^*_{(0,B_2)} \geq 0$ and $\hat{p}^*_{(1,B_2)} \geq 0$, respectively.*

After the publication of Silva *et al.* [65], Kim and Kim [35] proved that Conjecture 1 holds. In the remainder of this section we provide numerical results that illustrate this point, and help understand the intuition behind this result.

Table 1: Breakdown of cases we studied.

Case	Description
A	$\lambda < \mu_1 < \mu_2$
B	$\lambda < \mu_2 < \mu_1$
C	$\mu_1 < \lambda < \mu_2$
D	$\mu_1 < \mu_2 < \lambda$
E	$\mu_2 < \lambda < \mu_1$
F	$\mu_2 < \mu_1 < \lambda$

We conduct numerical experiments considering a system where B_2 attains all integer values from 1 to 50, and also multiples of 50 up to 500. We generate 1000 sets of parameters independently and solve the problem numerically for each set with each value of the buffer size. In order to generate λ, μ_1, and μ_2, we sample from a continuous uniform distribution between 1 and 100, while c_1, c_2 are sampled from a continuous uniform distribution between 1 and 1000. If $c_1 \geq c_2$, we discard those cost values and generate a new set until $c_1 < c_2$. It is intuitive that if $c_1 > c_2$, then the Greedy policy is optimal, because incurring c_1 instead of risking the possibility of incurring a lower c_2 does not make financial sense. In Chapter 4, we prove this formally in Proposition 12. In order to analyze the results we divided the data sets into six cases as explained in Table 1.

Figure 4 illustrates how the average $p(0, B_2)$ changes with respect to B_2 for each of the cases above. Figure 5 and Figure 6 show the average $\hat{p}_{(0,B_2)}$ and $\hat{p}_{(1,B_2)}$ respectively, for different values of B_2. All probability values in these figures were calculated as averages for a fixed buffer size over the instances described above. The half-width of the confidence interval for each point in Figures 4, 5 and 6 is less than 0.02 for cases E and F; for the remaining cases it is less than 0.01 and vanishes as B_2 increases. Our observations show that the probability of being in the states of interest decreases dramatically as a function of the buffer size until it practically vanishes by $B_2 = 10$ for cases A and C and by $B_2 = 50$ for cases B and D but never vanishes for cases E

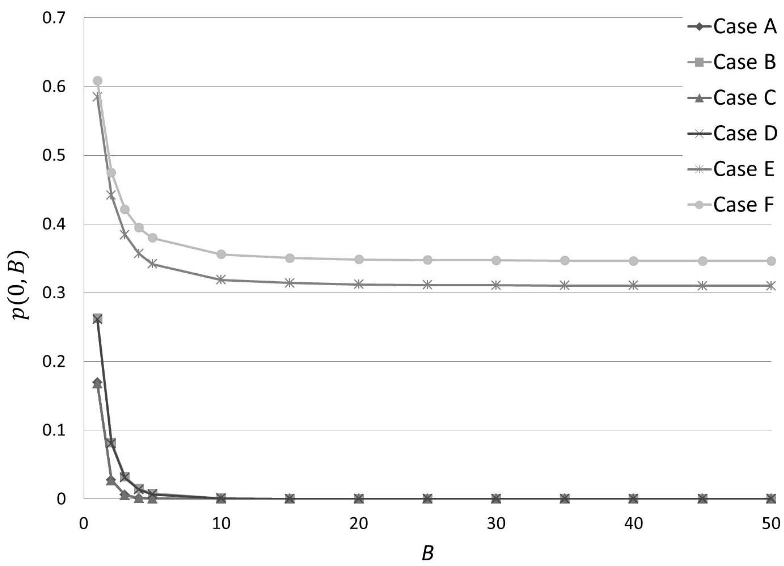

Figure 4: Average $p(0, B_2)$ for increasing values of B_2 for cases A-F

and F (i.e., when μ_2 is smaller than μ_1 and λ). For cases E and F, as B_2 increases, all probabilities of interest monotonically decrease and eventually converge to a strictly positive value.

These results suggest that choosing a non-optimal policy (amongst the Prudent and Greedy policies) may not have a significant impact on the cost when parameters satisfy cases A-D, but might have an impact when they satisfy cases E and F. In particular, we can conjecture that $c_*(10)$ can be a good approximation for $c_*(B_2)$ for $B_2 > 10$ in cases A to D, however, the same may not hold in cases E and F. Thus, it is important to develop heuristic policies that yield good cost performance for larger values of B_2 under these scenarios as the risk remains that we incur unnecessary costs by choosing the wrong policy.

3.3.3 Heuristics and Numerical Experiments

As we have shown, the optimal policy is characterized by a threshold $c_*(B_2)$. In the previous section we saw that for large values of B_2, the probabilities of being in a full state vanish quickly as B_2 increases for all systems where $\mu_2 > \lambda$ and/or

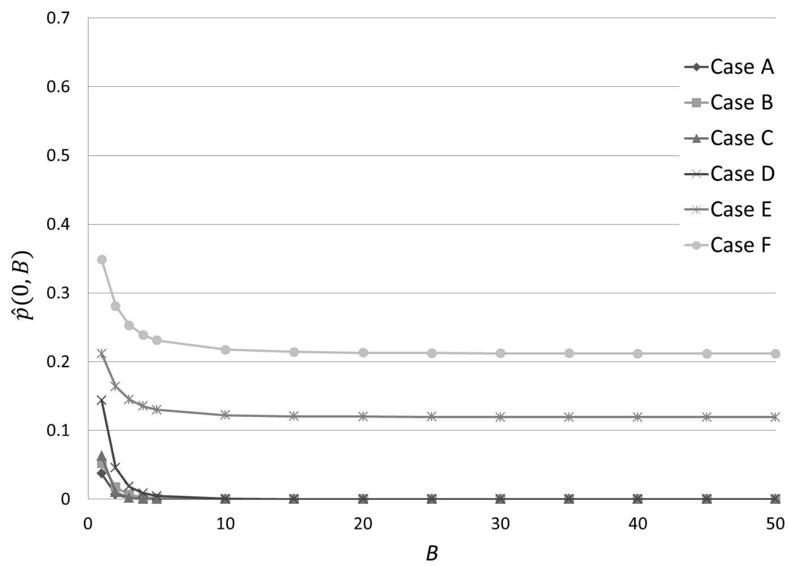

Figure 5: Average $\hat{p}(0, B_2)$ for increasing values of B_2 for cases A-F

$\mu_2 > \mu_1$ but they are not negligible otherwise (namely cases E and F). In this section we propose three easily implementable heuristic policies and show through numerical experiments that they can yield near-optimal long-run average cost performance for all buffer sizes.

3.3.3.1 Descriptions of the heuristics

The first heuristic uses the following upper bound.

Proposition 5. *The following upper bound holds for* $1 \leq B_2 \leq 10$

$$\hat{c}(B_2) = \left(1 + \frac{\mu_2}{\mu_1 + \lambda}\right) > c_*(B_2).$$

Proof. For the case $B_2 = 1$ we have from equation (3.3.1) that

$$c_*(1) = \left(1 + \frac{\mu_2^2}{\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2}\right) = \left(1 + \frac{\mu_2}{\frac{\lambda\mu_1}{\mu_2} + \lambda + \mu_1}\right) < \left(1 + \frac{\mu_2}{\mu_1 + \lambda}\right) = \hat{c}.$$

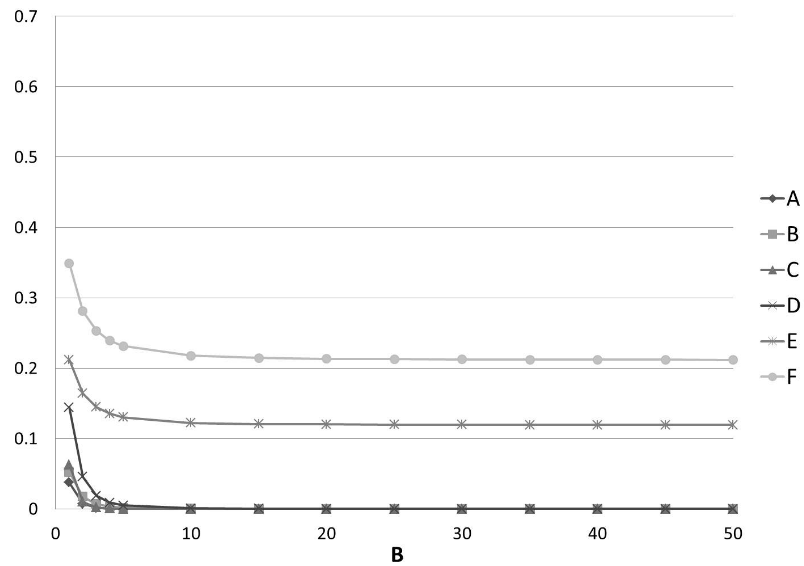

Figure 6: Average $\hat{p}(1, B_2)$ for increasing values of B_2 for cases A-F

Similarly for $B_2 = 2$ we have from equation (3.3.1)

$$\begin{aligned}
c_*(2) &= \left(1 + \frac{\mu_2^3\left(\lambda + \mu_1 + \mu_2\right)}{\mu_1\mu_2^2(\mu_1+\mu_2) + \lambda\mu_2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right)}\right) \\
&< \left(1 + \frac{\mu_2^3\left(\lambda + \mu_1 + \mu_2\right)}{\mu_1\mu_2^2(\mu_1+\mu_2) + 2\lambda\mu_1\mu_2^2 + \lambda^2\mu_2^2 + \lambda\mu_2^3}\right) \\
&= \left(1 + \frac{\mu_2}{\mu_1 + \lambda}\right) = \hat{c}
\end{aligned}$$

Note that the first inequality holds because $2\lambda\mu_1\mu_2^2 + \lambda^2\mu_2^2 + \lambda\mu_2^3 < \lambda\mu_2(\mu_1+\mu_2)^2 + \lambda^2\left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right)$ as all parameters are positive. We can prove the inequality for all values of $B_2 \leq 10$, using the values of $c_*(B_2)$ in the Appendix and following a similar procedure. ■

We further conjecture that this bound holds for any value of $B_2 < \infty$. To understand how the bound relates to the actual value $c_*(B_2)$, we calculated the exact $c_*(B_2)$ and the bound for the data sets from the experiments presented in Section 3.3.2. Our numerical experiments suggest that $\hat{c}(B_2)$ is indeed an upper bound for all the values of $B_2 > 10$ considered. Furthermore, we observe that the bound is much tighter for cases A-D than cases E and F. This leads us to consider a heuristic that uses $\hat{c}(B_2)$ as a proxy for $c_*(B_2)$. However, since the bound is not as tight for cases E and F, one may need to be more prudent if system parameters satisfy the assumptions of cases E and F. We would like to point out that the numerical experiments illustrated that $c_*(B_2)$ is not monotone in B_2. Our results and observations from numerical experiments lead to the following three heuristics:

- **Heuristic 1:** if $\frac{c_2}{c_1} > \hat{c}(B_2)$ use the Prudent policy, else use the Greedy policy.
- **Heuristic 2:** if the parameters satisfy cases A-D use Heuristic 1, otherwise, use the Prudent policy.
- **Heuristic 3:** if $B_2 \leq 10$, calculate $c_*(B_2)$ and use the optimal policy, otherwise, if $B_2 > 10$ and $\frac{c_2}{c_1} > c_*(10)$ use the Prudent policy, else if $B_2 > 10$ and $\frac{c_2}{c_1} \leq c_*(10)$ use the Greedy policy.

Heuristics 1 and 2 are simple and easily implementable. Heuristic 3 requires more computational effort than the other two heuristics since one needs to compute $c_*(B_2)$ for all $B_2 \leq 10$, but can still be implemented in a simple spreadsheet. Note that Heuristic 1 chooses the Greedy policy when it is optimal since $c_*(B_2) > \frac{c_2}{c_1}$ implies that $\hat{c}(B_2) > \frac{c_2}{c_1}$. However, if the Prudent policy is optimal it does not necessarily choose the optimal policy. The same is true about Heuristic 2 in cases A-D but the opposite holds in cases E and F. In these two cases, Heuristic 2 always chooses right if the Prudent policy is optimal but will not choose the optimal policy if the Greedy policy is optimal. For Heuristic 3, since we do not have a monotonicity result for $c_*(B_2)$, we cannot predict which policy will be favored. However, since $c_*(B_2)$ appears to change very slowly in B_2, using $c_*(10)$ might yield a better long-run average cost performance than using $\hat{c}(B_2)$.

3.3.3.2 Numerical results from heuristic policies

For the data sets described in Section 3.3.2, we calculate both the optimal long-run average cost and the long-run average cost under each heuristic. We compare the cost (not the gain), as this is a more intuitive, yet equivalent measure. We compare the proposed heuristics to an alternative of always choosing the Greedy policy or always using the Prudent policy. Tables 2 and 3 summarize the results for average cost performance for cases A to F. Each column shows the average percentage of additional cost incurred by the policy over the optimal cost, with its corresponding confidence interval. On the other hand, Table 4 shows the results for the worst additional cost incurred over the optimal cost.

Table 2: Average percentage of excess over the optimal cost, under the Greedy and Prudent policies

Case	Greedy	Prudent
A	0.022±6.3E-3%	0.123±2.3E-2%
B	0.412±5.8E-2%	0.072±1.7E-2%
C	0.000±1.6E-4%	0.042±6.9E-3%
D	0.011±2.2E-3%	0.023±4.2E-3%
E	2.517±5.9E-2%	0.041±6.6E-3%
F	1.524±3.8E-2%	0.023±4.2E-3%
Total	0.776±1.7E-2%	0.053±5.0E-3%

Table 3: Average percentage of excess over the optimal cost, under the heuristic policies

Case	Heuristic 1	Heuristic 2	Heuristic 3
A	0.003±1.3E-3%	0.003±1.3E-3%	0.000±0.0E+0%
B	0.007±2.4E-3%	0.007±2.4E-3%	0.000±0.0E+0%
C	0.000±1.6E-4%	0.000±1.6E-4%	0.000±0.0E+0%
D	0.004±1.4E-3%	0.004±1.4E-3%	0.000±0.0E+0%
E	0.593±2.3E-2%	0.041±6.6E-3%	0.000±1.0E-4%
F	0.418±1.5E-2%	0.023±4.2E-3%	0.000±1.7E-5%
Total	0.179±5.1E-3%	0.013±1.4E-3%	0.000±1.8E-5%

The results of Tables 2 and 3 illustrate that Heuristic 3 outperforms the other two heuristics. Clearly, the Greedy policy is the worst as the average additional cost of choosing the Greedy policy when it is not optimal is far larger than the average additional cost of choosing the Prudent policy when it is not optimal. The next best policy is Heuristic 1, which improves considerably over the Greedy policy in all cases. However, it is interesting to note that on average the Prudent policy performs better than Heuristic 1 (in particular for cases E and F). Heuristic 2 takes the best results from Heuristic 1 and from the Prudent policy and gives better results. Finally, Heuristic 3 outperforms all other policies, showing that $c_*(10)$ is a very good proxy for $c_*(B_2)$, when $B_2 > 10$ (providing near optimal results).

It is clear from the results of Tables 2 and 3 that on the average all policies yield good long-run average cost performance. However, Table 4, which shows the largest percentage of additional cost incurred by each policy over the optimal cost, illustrates that one needs to be careful since the Greedy policy and the Prudent policy can yield poor long-run average cost performance for certain systems. The worst case performance of Heuristics 1, 2, and 3 are significantly better than the Greedy policy and the Prudent policy. In particular, Heuristic 3 has near-optimal performance in all cases.

Table 4: Worst case performances for each policy

Case	Greedy	Prudent	Heuristic 1	Heuristic 2	Heuristic 3
A	10.37%	25.97%	2.75%	2.75%	0.00%
B	58.85%	25.81%	4.53%	4.53%	0.00%
C	0.67%	7.08%	0.67%	0.67%	0.00%
D	3.27%	5.44%	2.61%	2.61%	0.00%
E	12.38%	6.81%	5.40%	6.81%	0.08%
F	9.03%	5.73%	3.39%	5.73%	0.02%

Our results indicate that computing $c_*(B_2)$ for $B_2 \leq 10$ is sufficient and one can use Heuristic 3 to obtain near-optimal long-run average cost performance.

3.4 Conclusions

Some communication networks, such as the Internet, can be modeled as loss systems. These are queueing networks where each station has a finite capacity, and if a customer in the network encounters a station that is full, then (rather than blocking the previous station) the customer leaves the network. In this chapter, we study an admission control problem for a Markovian loss system consisting of two finite capacity service stations in tandem. Customers arrive to station 1 according to a Poisson process and a gatekeeper who has complete knowledge of the number of customers at both stations decides to accept or reject each arriving customer. If a customer is rejected, a rejection cost c_1 is incurred. If an admitted customer finds that station 2 is full at the time of his service completion at station 1, he leaves the system and a loss cost c_2 is incurred. We used uniformization to model this system as a Markov decision process. We introduced two special policies. The Prudent policy, which only admits an arrival if the loss probability is zero; and the Greedy policy, which admits an arrival as long as there is space at the first station.

In this chapter, we analyzed the special case of having a buffer of size one at one station and an arbitrary finite buffer at the other. First, we reviewed the results of Zhang and Ayhan [75] for the case where $B_1 < \infty, B_2 = 1$. Then, for the remainder of the chapter, we focused on the case $B_1 = 1, B_2 < \infty$.

We proved that a policy which rejects arrivals when the buffer at the second station is not full cannot be optimal. Furthermore, we showed that the optimal policy has the structure that if $\frac{c_2}{c_1} > c_*(B_2)$ then the Greedy policy is optimal, otherwise the Prudent policy is optimal. We should note here that in this case the threshold on $\frac{c_2}{c_1}$ is a function of B_2, unlike the system studied by Zhang and Ayhan [75], where the threshold does not depend on the buffer size. We provided exact expressions for $c_*(B_2)$ for buffer values $1 \leq B_2 \leq 10$ and later showed that one does not need to compute $c_*(B_2)$ for larger values of B_2 to obtain near-optimal policies.

We also observed that, unless $\mu_2 < \min(\lambda, \mu_1)$, the probability of being in state $(0, B_2)$ under the Prudent policy and in states $(0, B_2)$ and $(1, B_2)$ under the Greedy policy, monotonically converge to 0 as B_2 increases. Similarly, these probabilities monotonically converge to a constant when μ_2 is the smallest rate. These observations on the probabilities led us to develop three heuristic policies with good long-run average reward performance. In particular, the heuristic that uses $c_*(10)$ as a proxy for $c_*(B_2)$ when $B_2 > 10$ results in near-optimal long-run average reward performance.

In the next chapter we consider the case where buffers at both stations are identical and of finite, arbitrary size. We will use the same MDP model developed in this chapter.

CHAPTER IV

ADMISSION CONTROL FOR A TANDEM LOSS SYSTEM WITH TWO STATIONS - GENERAL CASE

In this chapter, we continue to study the problem described in Chapter 3, but now we focus on the case where both stations have equal buffers of finite size. Specifically, we study the problem of admission control to a loss system comprised of two stations in tandem and one server at each station. Arrivals to the system follow a Poisson Process with rate λ and service times at each station follow an exponential distribution with rate μ_i, $i = 1, 2$. Each station has a finite capacity denoted by $B_i < \infty$, $i = 1, 2$. In this chapter, we focus on the case where $B_1 = B_2 = B$. Upon each arrival a gatekeeper has to decide (based on full knowledge of the state of the system) whether to admit or reject the arriving customer. If a customer is not admitted, a rejection cost c_1 is incurred. If a customer completes service at the first station and at that time the second station is full, the customer is lost and a loss cost of c_2 is incurred. Recall that if the first station is full at the time of an arrival, then the incoming customer has to be rejected and the rejection cost c_1 is incurred. The objective of the system administrator is to determine an admission control policy at the first station that minimizes the long-run average cost for the system.

Our goal, in this chapter, is to provide the system administrator with simple and easy-to-implement policies that will yield optimal or near-optimal performance in most cases. To do this, we start by analyzing the performance of the Prudent and Greedy policies, as well as establishing optimality conditions for each of them. Next, we show that there are some system states in which it is always optimal to admit incoming customers. Then, we fully characterize the optimal policy for the particular case of $B = 2$. We also use examples to illustrate that, in general, optimal policies may have a complicated structure. Finally, we propose two heuristic policies. The first heuristic uses only the two aforementioned policies, while the other uses our insights about the structure of the optimal policy. We use numerical experimentation to show that both of these heuristics yield near-optimal performance.

The remainder of this chapter is organized as follows: In Section 4.1, we focus on the Prudent policy, we characterize its long-run average reward, we provide a necessary and sufficient condition for optimality and we consider its limiting behavior as the buffer sizes go to infinity. In Section 4.2, we shift to the Greedy policy, we give a matrix-analytic characterization of its long-run average reward, we derive a sufficient condition for optimality and we also consider its limiting behavior. Section 4.3 is devoted to exploring the structure of the optimal policy. First, we identify states where it is always optimal to admit incoming customers; then we give a full characterization of the optimal policy when $B = 2$; and we end the section with a discussion on the complicated structure of optimal policies when $B \geq 3$. In Section 4.4, we introduce two heuristic policies and, through numerical experiments, show that these heuristics usually achieve near-optimal performance. Finally, Section 4.5 concludes the chapter. The results in this chapter can be found in Silva, Zhang and Ayhan [66].

For clarity of exposition, we present all the results in this chapter for the particular case of $B_1 = B_2 = B$. However, we note that all the results presented in Section 4.1 and Subsection 4.3.1 hold for any $B_1 \geq B_2$, because for a fixed B_2 the set of recurrent states under the Prudent policy is identical for any B_1 such that $B_1 \geq B_2$. Therefore, all the proofs in those sections follow immediately, substituting B for B_2. Furthermore, the analytical results in Section 4.2 hold for any general $B_1 \geq 1, B_2 \geq 1$. Since now we only have one buffer size parameter B, we simplify the notation for the gain under policy π as $g_\pi(B)$. Consequently, the gain under the Prudent policy is

$g_P(B)$, under the Greedy policy it is $g_G(B)$ and the optimal gain is $g^*(B)$.

4.1 The Prudent Policy

In this section, we explore the steady state behavior of the Prudent policy and give necessary and sufficient conditions for it to be optimal. We also consider the limiting behavior of the Prudent policy as $B \to \infty$. Consider a system with $B \geq 1$. Let us partition the state space in a way that will be useful for our analysis:

$$S_0 = \{s \in S : s = (q_1, q_2), q_1 + q_2 < B, q_1 < B\},$$

$$S_1 = \{s \in S : s = (q_1, q_2), q_1 + q_2 = B, q_1 < B\},$$

$$S_2 = \{s \in S : s = (q_1, q_2), q_1 + q_2 > B, q_1 < B\}, \text{ and}$$

$$S_3 = \{s \in S : s = (q_1, q_2), q_1 = B\}.$$

Note that $\{S_0, S_1, S_2, S_3\}$ is clearly a partition of S. Since this model is unichain, has finite state and action spaces and bounded rewards, there exists a stationary deterministic policy which is optimal. Let Π be the set of all stationary deterministic policies.

Let us generalize the definition of the Prudent policy, by considering a *Prudent Family* of policies. Define

$$\Pi_P = \left\{ \pi \in \Pi : \pi(s) = \begin{cases} 1 & s \in S_0 \\ 0 & s \in S_1 \\ \text{any action} & s \in S_2 \\ 0 & s \in S_3 \end{cases} \right\}.$$

Since the Markov chain under any policy $\pi \in \Pi_P$ has the same steady state behavior the following results apply to all such policies, not just π_P.

Proposition 6. *For any policy $\pi \in \Pi_P$, the long-run average reward will be given by*

$$g_P(B) = -c_1 \frac{\lambda}{G} \sum_{k=0}^{B} \frac{\lambda^B}{\mu_1^k \mu_2^{B-k}},$$

where

$$G = \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \left(\frac{\lambda}{\mu_1}\right)^{q_1} \left(\frac{\lambda}{\mu_2}\right)^{q_2}.$$

Proof. Consider the continuous-time MDP (before uniformization) defined in Section 3.1, under policy $\pi \in \Pi_P$. Note that for any policy $\pi \in \Pi_P$ the resulting continuous-time Markov chain has a single recurrent class given by $R = S_0 \cup S_1$ and transient states given by $T = (S_2 \cup S_3)$.

Now consider the closed Jackson network in Figure 7, where station A has independent identically distributed exponential service times with mean $1/\lambda$ and stations 1 and 2 are the same as in our model.

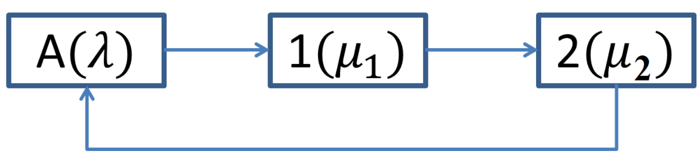

Figure 7: Closed Jackson network equivalent model.

If the network in Figure 7 has B units in it, let q_0 be the number of units in station A, then we can model its behavior as a continuous-time Markov chain (CTMC) with the following state space:

$$\left\{ X(t) = (q_0, q_1, q_2) : q_0, q_1, q_2 \in \mathbb{Z}_+, \sum_{k=1}^{3} q_k = B \right\}.$$

Note that if there are B units at station 1 and 2 combined, then there are 0 units at station A and hence no arrivals to station 1. More specifically $q_0 = B - q_1 - q_2$, hence we will only have the possibility of arrivals to station 1 when $q_1 + q_2 < B$. Furthermore, the arrival rate to station 1 is λ if $q_1 + q_2 < B$ and zero otherwise.

Therefore, this closed Jackson network and our model under any policy in Π_P are equivalent. In particular, the stationary distribution of $X(t)$ will be the same as the stationary distribution of the recurrent states for our model. Then using the classical result from Gordon and Newel [21], we get that if $\tilde{p}(s)$ is the stationary distribution in state s of the Jackson network CTMC model, then,

$$\tilde{p}(q_1, q_2) = \frac{\lambda^{q_1+q_2}}{\mu_1^{q_1}\mu_2^{q_2}} \times \frac{1}{G}.$$

Therefore, since any policy in the Prudent family of policies rejects new arrivals in states $s \in S_1$ and admits them in $s \in S_0$, and $S_2 \subset T$ then the long-run average reward of any policy in the Prudent family is:

$$g_P(B) = -c_1\lambda \sum_{k=0}^{B} \tilde{p}(k, B-k) = -c_1 \frac{\lambda}{G} \sum_{k=0}^{B} \frac{\lambda^B}{\mu_1^k \mu_2^{B-k}}.$$

■

4.1.1 Conditions for Optimality of the Prudent Policy

In the previous section, we derived a closed form expression for the long-run average reward under the Prudent policy. In this section, we use a linear programming (LP) approach to derive both necessary and sufficient conditions for the optimality of the Prudent policy. In particular, we solve a LP where each decision variable corresponds to a state-action pair of the Markov Decision process. We derive closed-form expressions for the reduced cost of each basic variable under the Prudent policy (each stationary deterministic policy corresponds to a basis of the LP polyhedron) and determine conditions for optimality by determining bounds for these reduced costs to be non-negative.

Recall the LP approach to solving MDPs (see e.g. Hordijk and Kallenberg [25]). It is well known that for a unichain MDP the solution to the following LP is a solution to the optimality equation (3) of the MDP,

$$\begin{aligned} \min \quad & g \\ s.t. \quad & g + h(s) - \sum_{j \in S} p(j|s,a)h(j) \geq r(s,a) \quad \forall a \in A_s, s \in S, \end{aligned} \tag{4}$$

where g represents the long-run average reward and $h(s)$ is the bias of state s as defined in (3). The dual of (4) is

$$\begin{aligned} \max & \sum_{s \in S} \sum_{a \in A_s} r(s,a)x(s,a) \\ s.t. & \sum_{s \in S} \sum_{a \in A_s} x(s,a) = 1 \\ & \sum_{a \in A_s} x(j,a) - \sum_{s \in S} \sum_{a \in A_s} p(j|s,a)x(s,a) = 0 \quad \forall j \in S \\ & x(s,a) \geq 0 \quad \forall a \in A_s, s \in S. \end{aligned} \tag{5}$$

Note that there is a redundant constraint in the dual LP. It is also known that for unichain MDP models there is a correspondence between the extreme points of the dual (5) and stationary deterministic policies, determined up-to recurrent states (see Kallenberg [31]). That is, multiple policies that have the same set of recurrent states and are equal in their recurrent states, but differ only in the actions taken in transient states correspond to the same extreme point in the dual polyhedron. A key part of this equivalence is that for unichain models, under a deterministic stationary policy π, the resulting Markov chain has a single recurrent class of states R_π and a unique stationary distribution $\bar{p}_\pi$. Namely, recall from Chapter 3 that we call the stationary distribution under the Prudent policy p, and under the Greedy policy $\hat{p}$. Then, the corresponding dual LP (5) has a basic feasible solution given by

$$x(s,a) = \begin{cases} \bar{p}_\pi(s) & s \in R_\pi, \pi(s) = a, \\ 0 & \text{otherwise.} \end{cases}$$

We should note that if there are transient states the above solution will correspond to a degenerate extreme point in the dual polyhedron. This means that multiple stationary deterministic policies may correspond to the same extreme point of the polyhedron. For instance,under for the Prudent policy π_P, there is a corresponding dual basic feasible solution given by

$$\bar{x}(s,a) = \begin{cases} p(s) & s \in S_0 \cup S_1, \pi_P(s) = a, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

However, since the MDP model is not recurrent, this extreme point is degenerate and corresponds to more than one basis. Specifically, every policy $\pi \in \Pi_P$ in the MDP has the same corresponding solution to the dual LP, with the same optimal value, albeit with different basis matrices.

Finally, let us mention that although one must run the policy iteration algorithm to completion to find a solution to equation (3), it has been shown that, in unichain models it is not necessary to run it to completion to find an optimal policy (see Puterman [58], Section 8.6.1). To show that a stationary deterministic policy is optimal it suffices to show (in the policy improvement step) that changing the action in recurrent states will not improve the long-run average reward. By the equivalence with LP, it follows that: if in any iteration of the simplex method applied to (5) the reduced costs of the non-basic variables corresponding to recurrent states are non-positive, then the policy (or policies) corresponding to the current solution is optimal. This is true regardless of the reduced costs for the variables corresponding to transient states.

We can apply the LP approach here to show necessary and sufficient conditions for π_P to be optimal. Starting the simplex algorithm for (5) with the solution corresponding to the Prudent policy given by (6) results in having the basis matrix $\boldsymbol{B}$ whose indices are given by $\{(s,a) : s \in S_0, a = 1\} \cup \{(s,a) : s \notin S_0, a = 0\}$. Therefore, the non basic matrix $\boldsymbol{N}$ will have the index set: $\{(s,a) : s \in S_0, a = 0\} \cup \{(s,a) : s \in S_1 \cup S_2, a = 1\}$. This way, there is a basic variable for each state and its corresponding action under π_P, and there is a non-basic variable for each state where $|A_s| = 2$

paired with the opposite action. Furthermore, since we have a redundant constraint we arbitrarily eliminate the constraint corresponding to state $(0,0)$ from (5).

Since for every state in S_3 there is only one available action, and every state in S_2 is transient under π_P, then, by the previous discussion, to show optimality of the Prudent policy it suffices to show that for every non-basic variable $\bar{x}(s,a), s \in S_0 \cup S_1$ that corresponds to a column of $\boldsymbol{N}$ the reduced costs are non-positive. That is

$$\bar{c}(s,a) = r(s,a) - \boldsymbol{c_B} \cdot \boldsymbol{B}^{-1} \cdot \boldsymbol{N}_{\cdot(s,a)} \leq 0$$

where $\boldsymbol{c_B}$, is the vector of costs of the basic variables. In this case,

$$\boldsymbol{c_B}(s,a) = \begin{cases} 0 & \{(s,a) : s \in S_0, a = 1\}, \\ -c_1\lambda & \{(s = (q_1,q_2),a) : s \notin S_0, q_2 < B, a = 0\}, \\ -c_1\lambda - c_2\mu_1 & \{(s = (q_1,q_2),a) : s \notin S_0, q_2 = B, a = 0\} \end{cases}$$

where $\boldsymbol{B}^{-1}$ is the inverse of the basis matrix. And $\boldsymbol{N}_{\cdot(s,a)}$ is the column of $\boldsymbol{N}$ that contains the coefficients on the left hand side of the dual LP problem (5) corresponding to state-action combination (s,a).

Proposition 7. *If we start the simplex algorithm for* (5) *with the solution corresponding to the Prudent policy given by* (6)*, for each non-basic variable of the form* $\bar{x}(s,0)$ *where* $s = (q_1,q_2) \in S_0$*, the reduced costs are given by*

$$\bar{c}(s,0) = -\left(\frac{c_1\lambda}{G}\right) \sum_{i=0}^{(B-1)-(q_1+q_2)} \left\{ \sum_{k=0}^{(B-1)-q_1} \left(\frac{\lambda}{\mu_1}\right)^k \left(\frac{\lambda}{\mu_1}\right)^i \right\}$$

where

$$G = \sum_{i=0}^{B} \sum_{j=0}^{B-i} \left(\frac{\lambda}{\mu_1}\right)^i \left(\frac{\lambda}{\mu_2}\right)^j.$$

Note that this means that, if $p(i,j)$ *is the long-run probability of being in state* (i,j) *under the Prudent policy* π_P *then we can rewrite the above as*

$$\bar{c}(s,0) = -(c_1\lambda) \sum_{i=0}^{(B-1)-(q_1+q_2)} \left\{ \sum_{k=0}^{(B-1)-q_1} p(i,j) \right\}.$$

Proof. The statement in the proposition can be verified algebraically for any B, by taking

$$\bar{c}(s,0) = -(c_1\lambda) - \boldsymbol{c_B} \cdot \boldsymbol{B}^{-1} \cdot \boldsymbol{N}_{\cdot(s,0)}$$

for each $(s,0)$ such that it corresponds to a column of $\boldsymbol{N}$ where $s \in S_0$. Note that the quantities c_B, $\boldsymbol{B}$ and $\boldsymbol{N}_{\cdot(s,0)}$ are all known. ■

Corollary 1. *If we start the simplex algorithm for* (5) *with the solution corresponding to the Prudent policy given by* (6)*, each non-basic variable* $\bar{x}(s,0)$ *is not a candidate to enter the basis whenever* $s \in S_0$*, because* $\bar{c}(s,0) < 0$*.*

It follows from Corollary 1 and the previous discussions that in order to prove the optimality of the Prudent policy it suffices to show that for every non-basic variable $\bar{x}(s,a)$ such that $s \in S_1, a = 1$, the reduced costs are non-positive.

Theorem 3. *Let*

$$c_*(B) = 1 + \frac{1}{G} \sum_{k=0}^{B-1} \sum_{r=0}^{k} \binom{k}{r} \sum_{n=1}^{B-r} \left(\frac{\lambda}{\mu_1}\right)^{n-1} \left(\frac{\mu_2}{\mu_1}\right)^{r+1}.$$

Then π_P *is an optimal policy if and only if* $\frac{c_2}{c_1} \geq c_*(B)$*.*

Proof. In order to verify whether π_P is optimal, we calculate the reduced costs for each $x(s,a)$ such that (s,a) has a corresponding column in $\boldsymbol{N}$, and s is recurrent under π_P i.e.

$$\bar{c}(s,a) = r(s,a) - \boldsymbol{c_B} \cdot \boldsymbol{B}^{-1} \cdot \boldsymbol{N}_{\cdot(s,a)} \leq 0.$$

By Corollary 1, it is sufficient to show that $\bar{c}(s,1) \leq 0$ for $s \in S_1$. It can be verified algebraically that for any B the reduced costs for $(s,1) : s = (q_1,q_2) \in S_1$ are given by,

$$\bar{c}((q_1,q_2),1) = -\lambda \left(\frac{\mu_1}{\mu_1+\mu_2}\right)^{q_1+1} \left(c_2 - c_1 - \frac{c_1}{G} \sum_{k=0}^{q_1} \sum_{r=0}^{k} \binom{k}{r} \sum_{n=1}^{B-k} \left(\frac{\lambda}{\mu_1}\right)^{n-1} \left(\frac{\mu_2}{\mu_1}\right)^{r+1}\right).$$

Hence, the condition for $\bar{c}((q_1,q_2),1) \leq 0$ reduces to

$$\frac{c_2}{c_1} \geq \left(1 + \frac{1}{G} \sum_{k=0}^{q_1} \sum_{r=0}^{k} \binom{k}{r} \sum_{n=1}^{B-k} \left(\frac{\lambda}{\mu_1}\right)^{n-1} \left(\frac{\mu_2}{\mu_1}\right)^{r+1}\right).$$

In the above expression, the right hand side is increasing in q_1 hence it is maximized when $q_1 = B - 1$, which is the same condition as $\frac{c_2}{c_1} \geq c_*(B)$. ■

The next result follows immediately,

Corollary 2. *Every policy $\pi \in \Pi_P$ is optimal if and only if $\frac{c_2}{c_1} \geq c_*(B)$.*

Proof. In Proposition 6, we showed that the long-run average reward for any $\pi \in \Pi_P$ is the same. Therefore, if one policy in the Prudent Family is optimal then they all are. ■

This is a necessary and sufficient condition for optimality of the Prudent policy. So, as $c_2 \to \infty$, or simply if c_2 is sufficiently greater than c_1, then the Prudent policy is optimal. This is logical, since if c_2 is large, the contribution of each loss to the long-run average cost becomes so high that it is better to reject any customer that might eventually be lost.

The condition in Theorem 3 determines when the Prudent policy is optimal from the system administrator's perspective. Spicer and Ziedins [68], considered the problem of having multiple tandem lines of two stations and finding a user-optimal policy, where the user's objective function is maximizing the probability of success (where success is defined as not being lost). In this case, because the system consists of only one line, the user has no choices to make. However, if every user's objective is to maximize the probability of success, then the Prudent policy is also user-optimal. This follows because the Prudent policy serves the greatest possible number of customers while guaranteeing that no customers are ever lost.

4.1.2 Limiting Behavior of the Prudent Policy

Now we consider the limiting behavior of the Prudent policy as the buffer B increases. First, we show that the long-run average gain under the Prudent policy is monotone increasing in B. To see the intuition behind this result, consider two systems under the prudent policy with the same parameters, except one system has a larger buffer.

The system with the larger buffer experiences rejections less often than the one with the smaller buffer. Since, under the prudent policy, costs are only incurred when rejections happen, then the larger system should have a greater long-run average gain.

Proposition 8. *The long-run average gain under the Prudent policy ($g_P(B)$) is an increasing function of B.*

Proof. Recall from Proposition 6 that

$$g_P(B) = -c_1\lambda \frac{\sum_{k=0}^{B} \frac{\lambda^B}{\mu_1^k \mu_2^{B-k}}}{\sum_{j=0}^{B} \sum_{i=0}^{B-j} \left(\frac{\lambda}{\mu_1}\right)^j \left(\frac{\lambda}{\mu_2}\right)^i}.$$

Multiply the numerator and denominator by $(\mu_1\mu_2)^B$ and re-arranging the sums we get:

$$\begin{aligned} g_P(B) &= -c_1\lambda \frac{\lambda^B \sum_{k=0}^{B} \mu_1^{B-k}\mu_2^k}{\sum_{j=0}^{B} \sum_{i=0}^{B-j} (\mu_1\mu_2)^B \left(\frac{\lambda}{\mu_1}\right)^j \left(\frac{\lambda}{\mu_2}\right)^i} \\ &= -c_1 \frac{\lambda^{B+1} \sum_{k=0}^{B} \mu_1^k \mu_2^{B-k}}{\sum_{j=0}^{B} \sum_{i=0}^{B-j} \lambda^j \mu_1^{B-j} \lambda^i \mu_2^{B-i}} \\ &= -c_1 \frac{\lambda^{B+1} \sum_{k=0}^{B} \mu_1^k \mu_2^{B-k}}{\sum_{j=0}^{B} \lambda^j \sum_{i=0}^{j} (\mu_1\mu_2)^{B-j} \mu_1^i \mu_2^{j-i}} \end{aligned}$$

Now we take the difference and get:

$$\begin{aligned} &g_P(B) - g_P(B+1) = \\ &= -c_1 \left(\frac{\lambda^{B+1} \sum_{k=0}^{B} \mu_1^k \mu_2^{B-k}}{\sum_{j=0}^{B} \lambda^j \sum_{i=0}^{j} (\mu_1\mu_2)^{B-j} \mu_1^i \mu_2^{j-i}} - \frac{\lambda^{B+2} \sum_{k=0}^{B+1} \mu_1^k \mu_2^{B+1-k}}{\sum_{j=0}^{B+1} \lambda^j \sum_{i=0}^{j} (\mu_1\mu_2)^{B+1-j} \mu_1^i \mu_2^{j-i}} \right) \end{aligned}$$

Consider the numerator of the difference in parenthesis above, we have

$$
\begin{aligned}
&\left(\lambda^{B+1}\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\right)\left(\sum_{j=0}^{B+1}\lambda^j\sum_{i=0}^{j}(\mu_1\mu_2)^{B+1-j}\mu_1^i\mu_2^{j-i}\right)\\
&\qquad-\left(\lambda^{B+2}\sum_{k=0}^{B+1}\mu_1^k\mu_2^{B+1-k}\right)\left(\sum_{j=0}^{B}\lambda^j\sum_{i=0}^{j}(\mu_1\mu_2)^{B-j}\mu_1^i\mu_2^{j-i}\right)\\
=&\lambda^{B+1}\left(\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\right)\left((\mu_1\mu_2)^{B+1}+\lambda(\mu_1\mu_2)^{B}(\mu_1+\mu_2)+...+\lambda^{B+1}\sum_{i=0}^{B+1}\mu_1^i\mu_2^{B+1-i}\right)\\
&-\lambda^{B+2}\left(\sum_{k=0}^{B+1}\mu_1^k\mu_2^{B+1-k}\right)\left((\mu_1\mu_2)^{B}+\lambda(\mu_1\mu_2)^{B-1}(\mu_1+\mu_2)+...+\lambda^{B}\sum_{i=0}^{B}\mu_1^i\mu_2^{B-i}\right)\\
=&\sum_{j=0}^{B}\left[\lambda^{B+1+j}(\mu_1\mu_2)^{B+1-j}\left(\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\sum_{i=0}^{j}\mu_1^i\mu_2^{j-i}-\sum_{k=0}^{B+1}\mu_1^k\mu_2^{B+1-k}\sum_{i=0}^{j-1}\mu_1^i\mu_2^{j-1-i}\right)\right]\\
=&\sum_{j=0}^{B}\left[\lambda^{B+1+j}(\mu_1\mu_2)^{B+1-j}\left(\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\sum_{i=0}^{j}\mu_1^i\mu_2^{j-i}\right.\right.\\
&\qquad\left.\left.-\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\sum_{i=1}^{j}\mu_1^i\mu_2^{j+1-i}-\mu_1^{B+1}\sum_{i=0}^{j-1}\mu_1^i\mu_2^{j-1-i}\right)\right]\\
=&\sum_{j=0}^{B}\left[\lambda^{B+1+j}(\mu_1\mu_2)^{B+1-j}\left(\sum_{k=0}^{B}\mu_1^k\mu_2^{B-k}\mu_1^j-\mu_1^{B+1}\sum_{i=0}^{j-1}\mu_1^i\mu_2^{j-1-i}\right)\right]\\
=&\sum_{j=0}^{B}\left[\lambda^{B+1+j}(\mu_1\mu_2)^{B+1-j}\left(\sum_{k=0}^{B-j}\mu_1^k\mu_2^{B-k}\mu_1^j\right.\right.\\
&\qquad\left.\left.+\sum_{k=B+1-j}^{B}\mu_1^k\mu_2^{B-k}\mu_1^j-\mu_1^{B+1}\sum_{i=0}^{j-1}\mu_1^i\mu_2^{j-1-i}\right)\right]\\
=&\sum_{j=0}^{B}\left[\lambda^{B+1+j}(\mu_1\mu_2)^{B+1-j}\left(\sum_{k=0}^{B-j}\mu_1^k\mu_2^{B-k-j}(\mu_1\mu_2)^j\right)\right]\\
=&(\lambda\mu_1\mu_2)^{B+1}\sum_{j=0}^{B}\left[\lambda^j\left(\sum_{k=0}^{B-j}\mu_1^k\mu_2^{B-k-j}\right)\right].
\end{aligned}
$$

Then the difference is

$$g_P(B) - g_P(B+1) =$$

$$- c_1 \frac{(\lambda\mu_1\mu_2)^{B+1} \sum_{j=0}^{B} \left[\lambda^j \left(\sum_{k=0}^{B-j} \mu_1^k \mu_2^{B-k-j}\right)\right]}{\left(\sum_{j=0}^{B} \lambda^j \sum_{i=0}^{j} (\mu_1\mu_2)^{B-j} \mu_1^i \mu_2^{j-i}\right)\left(\sum_{j=0}^{B+1} \lambda^j \sum_{i=0}^{j} (\mu_1\mu_2)^{B+1-j} \mu_1^i \mu_2^{j-i}\right)} < 0,$$

since all the parameters are strictly positive. Therefore, $g_P(B)$ is increasing in B. ■

Next we focus on the behavior of the long-run average gain of the Prudent policy $g_P(B)$ as $B \to \infty$. We find that $g_P(B)$ always converges to a limit. However, the limit depends on the relationships between the parameters as stated in the next proposition.

Proposition 9. *As the buffer size $B \to \infty$, the long-run average gain under the Prudent policy $g_P(B)$ converges as follows:*

1. *If $\lambda < \min(\mu_1, \mu_2)$, then $g_P(B) \to 0$.*
2. *If $\mu_2 < \min(\mu_1, \lambda)$, then $g_P(B) \to c_1(\mu_2 - \lambda)$.*
3. *If $\mu_1 < \min(\mu_2, \lambda)$, then $g_P(B) \to c_1(\mu_1 - \lambda)$.*

Proof.

1. We can re-write the numerator of $g_P(B)$ in Proposition 6 as:

$$-c_1\lambda \left[\left(\frac{\lambda}{\mu_2}\right)^B - \left(\frac{\lambda}{\mu_1}\right)^B \frac{\mu_2}{\mu_1}\right] \frac{\mu_1}{\mu_2 - \mu_1}$$

Clearly, since $\lambda < \mu_1$ and $\lambda < \mu_2$, as $B \to \infty$ this expression converges to 0. Similarly, we can re-write the denominator of $g_P(B)$ in Proposition 6 as:

$$\frac{\mu_2}{\mu_2 - \lambda}\left[\frac{\mu_1}{\mu_1 - \lambda}\left(1 - \left(\frac{\lambda}{\mu_1}\right)^{B+1}\right) - \frac{\mu_1}{\mu_1 - \mu_2}\left(\left(\frac{\lambda}{\mu_2}\right)^{B+1} - \left(\frac{\lambda}{\mu_1}\right)^{B+1}\right)\right]$$

Clearly, since $\lambda < \mu_1$ and $\lambda < \mu_2$, as $B \to \infty$ this expression converges to $\mu_1\mu_2/(\mu_2 - \lambda)(\mu_1 - \lambda)$. So $g_P(B) \to 0$.

2. We can reorganize terms in the expression for $g_P(B)$ above and get

$$g_P(B) = \frac{\frac{-c_1\lambda\mu_1}{\mu_1-\mu_2}\left[1-\left(\frac{\mu_2}{\mu_1}\right)^{B+1}\right]}{\frac{\lambda}{\mu_2-\lambda}\left[\frac{\mu_1}{\mu_1-\lambda}\left(\left(\frac{\mu_2}{\lambda}\right)^{B+1}-\left(\frac{\mu_2}{\mu_1}\right)^{B+1}\right)-\frac{\mu_1}{\mu_1-\mu_2}\left(1-\left(\frac{\mu_2}{\mu_1}\right)^{B+1}\right)\right]}$$

Now, since $\mu_2 < \mu_1$ and $\mu_2 < \lambda$, taking the limit as $B \to \infty$ we get

$$\begin{aligned}\lim_{B\to\infty} g_P(B) =& \frac{\frac{-c_1\lambda\mu_1}{\mu_1-\mu_2}}{\frac{\lambda}{\mu_2-\lambda}\left[-\frac{\mu_1}{\mu_1-\mu_2}\right]}\\ =& c_1(\mu_2-\lambda)\end{aligned}$$

which is less than 0 under our assumptions.

3. We can reorganize terms in the expression for $g_P(B)$ above and get

$$g_P(B) = \frac{\frac{-c_1\lambda\mu_1}{\mu_1-\mu_2}\left[\left(\frac{\mu_1}{\mu_2}\right)^{B}-\frac{\mu_2}{\mu_1}\right]}{\frac{\lambda\mu_2}{\mu_1(\mu_2-\lambda)}\left[\frac{\mu_1}{\mu_1-\lambda}\left(\left(\frac{\mu_1}{\lambda}\right)^{B+1}-1\right)-\frac{\mu_1}{\mu_1-\mu_2}\left(\left(\frac{\mu_1}{\mu_2}\right)^{B+1}-1\right)\right]}$$

Now, since $\mu_1 < \mu_2$ and $\mu_1 < \lambda$, taking the limit as $B \to \infty$ we get

$$\begin{aligned}\lim_{B\to\infty} g_P(B) =& \frac{\frac{-c_1\lambda\mu_1}{\mu_1-\mu_2}\left[\frac{-\mu_2}{\mu_1}\right]}{\frac{\lambda\mu_2}{\mu_1(\mu_2-\lambda)}\left[\frac{-\mu_1}{\mu_1-\lambda}+\frac{\mu_1}{\mu_1-\mu_2}\right]}\\ =& \frac{\frac{c_1\lambda\mu_1\mu_2}{\mu_1(\mu_1-\mu_2)}}{\frac{\lambda\mu_1\mu_2}{\mu_1(\mu_2-\lambda)}\left[\frac{\mu_2-\lambda}{(\mu_1-\lambda)(\mu_1-\mu_2)}\right]}\\ =& c_1(\mu_1-\lambda)\end{aligned}$$

which is less than 0 under our assumptions.

■

When $B = \infty$ the system is a two-station open Jackson network. In this case, the two station tandem queue is only stable if $\lambda < \min(\mu_1, \mu_2)$. Our results show that in that case $g_P(B) \to 0$, because in a stable system with infinite capacity rejections never occur. In this case, we also conclude that as $B \to \infty$ the gain of the Prudent policy approaches the optimal gain, as the optimal gain is bounded by zero. So, for very large systems where $\lambda < \min(\mu_1, \mu_2)$ the Prudent policy will be optimal, or close to optimal. However, in the remaining cases if $B = \infty$ then the system is not stable. In these cases, the gain converges to a negative constant, as $B \to \infty$, because the number of customers in the system will also go to infinity and hence the system will accumulate rejection costs, proportional to the rate that the number of customers goes to infinity.

In summary, in this section we proved the following results about the Prudent policy: the long-run average gain for the Prudent policy $g_P(B)$ is given by Proposition 6, which is a strictly increasing function of the buffer size B; we showed this gain converges either to zero or a negative constant, depending on the relations between the arrival rate and the service rates at each station; and, in Theorem 3 we gave necessary and sufficient conditions for the optimality of the Prudent policy. In the next sub-section we explore properties of the Greedy policy.

4.2 The Greedy Policy

In this section, we give a matrix-analytic solution for the stationary distribution under the Greedy policy. Clearly, if the stationary distribution is known, the long-run average reward can be computed directly. We also provide a necessary condition for the Greedy policy to be optimal, and for systems with $B \leq 2$ we give necessary and sufficient conditions for optimality of the Greedy policy.

As mentioned in Section 3.1, under a given policy the system can be modeled as a continuous time Markov chain with state space $S = \{(q_1, q_2) \in \mathbb{Z}^2 : 0 \leq q_1 \leq B, 0 \leq q_2 \leq B\}$. The transition rate diagram for the underlying continuous time Markov chain when the system operates under the Greedy policy π_G for the case with $B = 3$ is given in Figure 8.

Looking closely at Figure 8, we observe that this CTMC has the structure of a

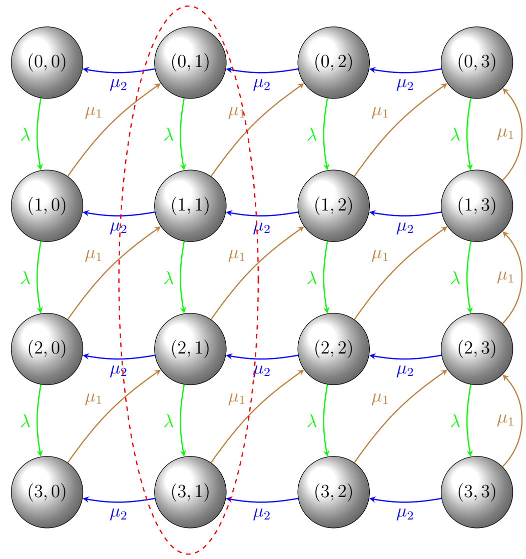

Figure 8: CTMC Under the Greedy Policy when $B = 3$.

finite Quasi Birth and Death process (QBD). For the specific example of Figure 8, we observe that there are four levels. We have circled one of the levels in Figure 8 with a dotted line to point it out. It is made up of all the states of the form $(q_1, 1), 0 \leq q_1 \leq B$. We can see that we will always have $B + 1$ levels (which for convenience we'll refer to as the zero-th, first, second, etc. up to B-th levels). It is clear from the structure of the Markov chain that all levels will contain the same number of phases and there will be $B + 1$ phases in each level.

The generator matrix for this continuous time Markov chain, denoted by $\boldsymbol{Q}$, is given as

$$
\boldsymbol{Q} = \begin{bmatrix}
\tilde{\boldsymbol{D}} & \boldsymbol{F} & & & & & \\
\boldsymbol{R} & \boldsymbol{D} & \boldsymbol{F} & & & & \\
 & \boldsymbol{R} & \boldsymbol{D} & \boldsymbol{F} & & & \\
 & & \cdot & \cdot & \cdot & & \\
 & & & \boldsymbol{R} & \boldsymbol{D} & \boldsymbol{F} & \\
 & & & & \boldsymbol{R} & \boldsymbol{D} & \boldsymbol{F} \\
 & & & & & \boldsymbol{R} & \hat{\boldsymbol{D}}
\end{bmatrix}
$$

where $\boldsymbol{R}$, $\boldsymbol{F}$, $\boldsymbol{D}$, $\tilde{\boldsymbol{D}}$ and $\hat{\boldsymbol{D}}$ are size $(B+1)$ square matrices. Suppose the row and column indices on each of these matrices are $i, j \in \{0, 1, ..., B\}$. Then, the elements of each sub-matrix are given by

$$
\boldsymbol{R}_{i,j} = \begin{cases} \mu_2 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \quad ; \quad \boldsymbol{F}_{i,j} = \begin{cases} \mu_1 & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases}
$$

$$
\boldsymbol{D}_{i,j} = \begin{cases} \lambda & \text{if } j = i+1 \\ -(\lambda + \mu_2) & \text{if } j = i = 0 \\ -(\lambda + \mu_1 + \mu_2) & \text{if } j = i, 1 \leq i \leq B-1 \\ -(\mu_1 + \mu_2) & \text{if } j = i = B \\ 0 & \text{otherwise} \end{cases}
$$

$$
\tilde{\boldsymbol{D}}_{i,j} = \begin{cases} \lambda & \text{if } j = i+1 \\ -\lambda & \text{if } j = i = 0 \\ -(\lambda + \mu_1) & \text{if } j = i, 1 \leq i \leq B-1 \\ -\mu_1 & \text{if } j = i = B \\ 0 & \text{otherwise} \end{cases}
$$

$$\hat{\boldsymbol{D}}_{i,j} = \begin{cases} \lambda & \text{if } j = i+1 \\ \mu_1 & \text{if } j = i-1 \\ -(\lambda+\mu_2) & \text{if } j = i = 0 \\ -(\lambda+\mu_1+\mu_2) & \text{if } j = i, 1 \leq i \leq B-1 \\ -(\mu_1+\mu_2) & \text{if } j = i = B \\ 0 & \text{otherwise} \end{cases}$$

Quasi Birth and Death processes have been studied in the literature (see for example Latouche and Ramaswami [42] and references therein). While finite QBD's have received less attention, a few methods for solving such problems have appeared in the literature. In particular, deNitto Persone and Grassi [15] propose an explicit matrix-analytic solution. Here we apply their method to the finite QBD described above. This method is based on building a recursion, so it is necessary to have enough levels for the definitions to make sense. Therefore, we require $B \geq 4$ to guarantee that the resulting finite QBD will have at least five levels. Note that for smaller values of B one can easily compute the stationary distribution using traditional methods.

Let $\boldsymbol{\eta} = [\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, ... \boldsymbol{\eta}_B]$ be the stationary distribution of the Markov chain, where $\boldsymbol{\eta}_k$ is the vector of steady state probabilities of the states in level k. Note that if $\hat{p}$ is the stationary distribution of the Markov chain under π_G, then $\eta_k(q_1) = \hat{p}(q_1, k)$ for all $k = 0, \ldots, B$ and $q_1 = 0, \ldots, B$. Then, the next result follows directly from Theorem 1 in deNitto Persone and Grassi [15].

Proposition 10. *Suppose $B \geq 4$ and define $\boldsymbol{\eta}$ as above, then*

$$\boldsymbol{\eta}_k = -\boldsymbol{\eta}_0 \frac{1}{\mu_2} \boldsymbol{F} \boldsymbol{C}_{11}^{k-2} + \boldsymbol{\eta}_1 \left(\boldsymbol{C}_{21}^{k-2} - \frac{1}{\mu_2} \boldsymbol{D} \boldsymbol{C}_{11}^{k-2} \right) \quad 2 \leq k \leq B-1$$

where $\boldsymbol{C}_{11}^{k-2}$ and $\boldsymbol{C}_{21}^{k-2}$ are $(B+1) \times (B+1)$ submatrices of the $2(B+1) \times 2(B+1)$

power matrix

$$\boldsymbol{C}^{k-2} = \begin{bmatrix} \boldsymbol{C}_{11}^{k-2} & \boldsymbol{C}_{12}^{k-2} \\ \boldsymbol{C}_{21}^{k-2} & \boldsymbol{C}_{22}^{k-2} \end{bmatrix}$$

where

$$\boldsymbol{C} = \begin{bmatrix} -\frac{1}{\mu_2}\boldsymbol{D} & \boldsymbol{I} \\ -\frac{1}{\mu_2}\boldsymbol{F} & \boldsymbol{0} \end{bmatrix}$$

with the convention that $\boldsymbol{C}^0 = \boldsymbol{I}$.

This determines $\boldsymbol{\eta}_k, \forall k : 2 \leq k \leq B-1$ in terms of $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$. In order to determine the vector $[\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \boldsymbol{\eta}_B]$ we use the next result, which follows directly from Corollary 1 of deNitto Persone and Grassi [15].

Proposition 11. *Suppose* $B \geq 4$ *and define* $\boldsymbol{\eta}$ *as above, then*

$$[\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \boldsymbol{\eta}_B] = [1, \boldsymbol{0}]\,(\boldsymbol{M}^*)^{-1}$$

where $[1, \boldsymbol{0}]$ *is a row vector of length* $3(B+1)$ *with a 1 in the first position and zeros in the rest. And* $\boldsymbol{M}^*$ *is the matrix resulting from taking the following matrix:*

$$\boldsymbol{M} = \begin{bmatrix} \tilde{\boldsymbol{D}} & -\frac{1}{\mu_2}\boldsymbol{F}\left(\boldsymbol{C}_{11}^{B-4}\boldsymbol{F} + \boldsymbol{C}_{11}^{B-3}\boldsymbol{D}\right) & -\frac{1}{\mu_2}\boldsymbol{F}\boldsymbol{C}_{11}^{B-3}\boldsymbol{F} \\ \boldsymbol{R} & \tilde{\boldsymbol{C}} & \left(\boldsymbol{C}_{21}^{B-3} - \frac{1}{\mu_2}\boldsymbol{D}\boldsymbol{C}_{11}^{B-3}\right)\boldsymbol{F} \\ \boldsymbol{0} & \boldsymbol{R} & \hat{\boldsymbol{D}} \end{bmatrix}$$

where

$$\tilde{\boldsymbol{C}} = \left(\boldsymbol{C}_{21}^{B-4} - \frac{1}{\mu_2}\boldsymbol{D}\boldsymbol{C}_{11}^{B-4}\right)\boldsymbol{F} + \left(\boldsymbol{C}_{21}^{B-3} - \frac{1}{\mu_2}\boldsymbol{D}\boldsymbol{C}_{11}^{B-3}\right)\boldsymbol{D}$$

and substituting the first column of $\boldsymbol{M}$ *by the following column vector:*

$$\begin{pmatrix} \left(\boldsymbol{I} - \sum_{k=2}^{B-1} -\frac{1}{\mu_2}\boldsymbol{F}\boldsymbol{C}_{11}^{k-2}\right)\boldsymbol{e} \\ \left(\boldsymbol{I} - \sum_{k=2}^{B-1}\left(\boldsymbol{C}_{21}^{k-2} - \frac{1}{\mu_2}\boldsymbol{D}\boldsymbol{C}_{11}^{k-2}\right)\right)\boldsymbol{e} \\ \boldsymbol{e} \end{pmatrix}$$

where $\boldsymbol{e}$ *is a vector of all ones.*

With this, we have a matrix-analytic solution for the stationary distribution under the Greedy policy. This is specially practical for large systems, where the number of equations that need to be solved to find the stationary distribution directly, that is $(B+1)^2$, is much greater than the size of the matrix $\boldsymbol{M}$, that is $3(B+1)$. Knowing the stationary distribution, the long-run average reward under the Greedy policy can be calculated as:

$$g_G(B) = -c_1\lambda \sum_{i=0}^{B} \eta_i(B) - c_2\mu_1 \sum_{i=1}^{B} \eta_B(i). \tag{7}$$

In Section 4.4, we use this result, together with the closed form solution for the stationary distribution under the Prudent policy to propose heuristics which yield near-optimal performance and require much less computational effort than solving for the optimal policy.

4.2.1 Conditions for Optimality of the Greedy Policy

In order to provide a sufficient condition on the optimality of the Greedy policy, we analyze the corresponding finite-horizon MDP model under the expected total reward criterion. Consider the MDP model we have been studying, but now assume a total of N periods. Let $v_{n,\pi}(\boldsymbol{s})$ be the expected total reward with n periods remaining, under policy π if the initial state is $\boldsymbol{s}$ and $v_n(\boldsymbol{s})$ be the optimal n-period reward with the initial state $\boldsymbol{s}$, or $v_n(\boldsymbol{s}) = \inf_\pi v_{n,\pi}(\boldsymbol{s})$. Here a policy is defined as a vector of functions $\pi = (\pi_N, \pi_{N-1}, ..., \pi_0)$, where each π_n is a map from the states into the action space, that is $\pi_n : S \to \{0, 1\}$.

The optimality equation is as follows

$$\begin{aligned} v_n(q_1, q_2) =& \mu_1[-c_2\mathbf{1}_{\{q_1>0 \text{ and } q_2=B\}} + v_{n-1}((q_1-1)^+, q_2 + \mathbf{1}_{\{q_1>0 \text{ and } q_2<B\}})] \\ &+ \mu_2 v_{n-1}(q_1, (q_2-1)^+) \\ &+ \lambda \max\{-c_1 + v_{n-1}(q_1, q_2), v_{n-1}(q_1+1, q_2)\}, \quad \forall (q_1, q_2) \in S. \end{aligned}$$

The optimal policy π^*, would take the following form: at state $s = (q_1, q_2) \in S_d$

with n periods to go

$$\pi_n^*(s) = \begin{cases} 0, & \text{if } v_{n-1}(q_1+1,q_2) < -c_1 + v_{n-1}(q_1,q_2) \\ 1, & \text{if } v_{n-1}(q_1+1,q_2) \geq -c_1 + v_{n-1}(q_1,q_2). \end{cases} \tag{8}$$

The following lemma is intuitive.

Lemma 1. *$\forall n \in \mathbb{Z}^+$,$v_n(q_1+1,q_2) \geq v_n(q_1,q_2) - (c_1 \vee c_2)$, $\forall q_1, q_2 \leq B$.*

Proof. Consider two systems with the same number of periods remaining in the decision horizon: system I under any policy π with the initial state (q_1, q_2), and system II with the initial state (q_1+1, q_2). Let system II operate under the same policy π as if the initial state were (q_1, q_2). Mark the last customer in the queue at station 1 in system II and always hold the marked customer as late as possible; specifically, serve the marked customer only if there is no one else at his station, and if there is an arrival to his station during his service, his service is preempted (in this case, his remaining service time, when he resumes service, is equal in distribution to his original service time by the exponential assumption). Also, if a rejection or a loss occur at the marked customer's station before he leaves the system, he is immediately unmarked and treated the same as all other customers afterwards.

With the two systems operating the aforementioned way, there are only two possibilities: (1) The marked customer leaves system II before any difference arises in the costs in these two systems; after that, both systems evolve identically and thus the same total cost is incurred in both systems by the end of the decision horizon. (2) One more rejection or loss occurs in system II than in system I at one of the two stations (where the marked customer is located at that moment), incurring an extra cost c_1 or c_2, and after that both systems evolve identically. In other words, by initially holding one more customer at station 1, system II either has the same total cost, or c_1 or c_2 more than system I. Because this is true for any policy π, the result follows. ■

Proposition 12. *If $c_1 \geq c_2$, the Greedy policy π_G is long-run average reward optimal.*

Proof. It follows from $c_1 \geq c_2$ and Lemma 1 that

$$v_n(q_1 + 1, q_2) \geq v_n(q_1, q_2) - c_1. \tag{9}$$

This, together with equation (8), then implies that the greedy policy π_G is optimal for the n-period problem under the expected total cost criterion, $\forall n \in \mathbb{Z}^+$. Taking the limit as $n \to \infty$ yields the result. ∎

This result is intuitive, as rejecting an arrival and incurring cost c_1 with certainty, rather than admitting it and taking the risk of incurring cost c_2 only makes sense if $c_1 < c_2$. Additionally, for small systems, we find that there exists a threshold that provides a necessary and sufficient condition for optimality of the Greedy policy.

Proposition 13. *Suppose $B = 1$ then the Greedy policy is optimal if and only if*

$$\frac{c_2}{c_1} \leq \left(1 + \frac{\mu_2^2}{\lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2}\right)$$

Suppose $B = 2$ then the Greedy policy is optimal if and only if

$$\frac{c_2}{c_1} \leq \left(1 + \frac{\mu_2^3\left(\mu_2^2(\lambda + \mu_1) + \mu_2(2\lambda + \mu_1)(\lambda + 2\mu_1) + (\lambda + \mu_1)^3\right)}{\beta_1}\right)$$

where

$$\begin{aligned}\beta_1 =& \lambda^2\mu_1^2(\lambda + \mu_1)^2 + \mu_2^4\left(\lambda^2 + \lambda\mu_1 + \mu_1^2\right) + 2\mu_2^3(\lambda + \mu_1)\left(\lambda^2 + \lambda\mu_1 + \mu_1^2\right) \\ &+ \mu_2^2(\lambda + \mu_1)^2\left(\lambda^2 + 3\lambda\mu_1 + \mu_1^2\right) + \lambda\mu_1\mu_2(\lambda + \mu_1)\left(\lambda^2 + 4\lambda\mu_1 + \mu_1^2\right)\end{aligned}$$

Proof. The proof for $B = 1$ follows from Theorem 2 in Chapter 3. The proof for $B = 2$ follows from the proof of Lemma 15 in Section 4.3.2. ∎

4.2.2 Limiting Behavior of the Greedy Policy

In order to understand the behavior of $g_G(B)$ as $B \to \infty$, we performed numerical experiments. We created 1000 randomly generated data sets. For each set we drew

λ, μ_1 and μ_2 independently from a continuous uniform distribution between 0 and 100, and drew c_1 and c_2 from a uniform distribution between 0 and 1000, discarding values of $c_2 < c_1$. For each data set, we calculated $g_G(B)$ (the gain under the Greedy policy), increasing B from 1 to 30.

We observe that if $\lambda < \min(\mu_1, \mu_2)$ then as B increases, $g_G(B) \to 0$. This is expected, because in this case as $B \to \infty$ the system approaches a two-station tandem open Jackson network. Therefore, if the system is stable at $B = \infty$, there are never any losses and $g_G(B) = 0$. So in the stable case, the gain of the Greedy policy approaches to the optimal gain as $B \to \infty$. If $\mu_1 < \min(\lambda, \mu_2)$, we observe that the gain $g_G(B)$ appears to be increasing and converges to a constant. On the other hand, if $\mu_2 < \min(\mu_1, \lambda)$, the gain $g_G(B)$ can be decreasing, increasing or even non-monotone. For this reason it is difficult to predict how the Greedy policy will perform in large systems, unless $\lambda < \min(\mu_1, \mu_2)$.

To summarize, in this section we provided the following results about the Greedy policy: the long-run average gain for the Greedy policy has a matrix-analytic solution; the Greedy policy is always optimal when $c_1 > c_2$; and, for small systems, there exists a threshold on $\frac{c_2}{c_1}$ that determines necessary and sufficient conditions for the optimality of the Greedy policy. In the next section, we show that the Prudent and Greedy policies are opposites in the sense that no policy can be optimal and admit fewer customers than the Prudent policy and, of course, no policy admits more customers than the Greedy policy.

4.3 Discussion on the Structure of the Optimal Policy

So far, we have established necessary and sufficient conditions for the Prudent policy to be optimal, and a necessary condition for the Greedy policy to be optimal. It remains to consider the case when neither of these policies is optimal. In this section, first, we show that every optimal policy must admit arrivals in the states where the

Prudent policy admits arrivals. Next, we focus on the special case where the buffer size B is equal to two. For this particular case, we fully characterize the optimal policy and observe that certain monotonicity properties hold. We then explore what happens when $B \geq 3$. We show through an example that some monotonicity properties that hold for $B = 2$, do not always hold for larger buffers.

4.3.1 It Is Never Optimal to Reject in States where Customers Cannot Be Lost

In this section we show that for each state in the set S_0 the optimal action is always to admit incoming arrivals. This means that any optimal policy must admit customers in the states where the Prudent policy admits arrivals.

Let $\boldsymbol{B}$ be the basis of the dual LP Problem (5), when the simplex method starts with the solution in (6). The rows of the basis $\boldsymbol{B}$, are indexed by the constraints of the dual LP, so there is one row per state, similarly the columns are indexed by the basic state-action combinations. Let $\boldsymbol{B}^{-1}$ be the inverse of this matrix and call its elements $b_{(s,a),s'}$, where (s, a) corresponds to a state-action combination in the basis $\boldsymbol{B}$ and $s' \in S$. Let $\boldsymbol{N}$ be the matrix of non-basic columns of the dual LP Problem (5) when starting the simplex method with (6). Call its elements $n_{s',(s,a)}$, where $s' \in S$ and (s, a) is a non-basic state-action combination with its corresponding column in $\boldsymbol{N}$.

Lemma 2. *Consider a state $s' = (q'_1, q'_2)$, then for each state-action pair in the set $\{(s, 0) : s \in S_2 \cup S_3\}$, where $s = (q_1, q_2)$ we have the following*

1. $b_{(s,0),s'} = 0$ *whenever* $q'_1 + q'_2 < q_1 + q_2$ *or* $q'_1 < q_1$.
2. $b_{(s,0),s'} = \frac{1}{\mu_1+\mu_2}$ *whenever* $q'_1 = q_1$ *and* $q'_2 < q_2$.
3. *For each* $s' \neq s$, *such that* $q'_1 + q'_2 \geq q_1 + q_2$ *and* $q'_1 \geq q_1$

$$b_{(s,0),s'} = \frac{1}{\mu_1 + \mu_2} \left(\mu_1 b_{(s,0),s^+} + \mu_2 b_{(s,0),s^-}\right)$$

where $s^+ = (q_1' - 1, q_2' + \mathbf{1}_{q_2' < B})$ *and* $s^- = (q_1', q_2' - 1)$.

Proof. These three statements can be verified algebraically for any buffer size by obtaining $\boldsymbol{B}^{-1}_{(s,0)\cdot}$ (the row of the inverse of the basis corresponding to state action combination $(s, 0)$) and formulating and solving a system of equations, by taking each basic variable $x(\hat{s}, a)$ and writing the equations:

$$\boldsymbol{B}^{-1}_{(s,0)\cdot} \cdot \boldsymbol{B}_{\cdot(\hat{s},a)} = \mathbf{1}_{s=\hat{s}}$$

where $\boldsymbol{B}_{\cdot(\hat{s},a)}$ is the column of the basis matrix corresponding to the basic variable $x(\hat{s}, a)$. ■

Furthermore, we can use the above to understand some entries in the simplex tableau for this problem. Define

$$m_{(s,a),(s',a')} = \boldsymbol{B}^{-1}_{(s,a)\cdot} \cdot \boldsymbol{N}_{\cdot(s',a')}$$

where $\boldsymbol{B}^{-1}_{(s,a)\cdot}$ is the row of the inverse of the basis corresponding to state-action combination (s, a) and $\boldsymbol{N}_{\cdot(s',a')}$ is the column of $\boldsymbol{N}$ corresponding to state-action combination (s', a').

Lemma 3. *For each* $(s, a) \in \{(s, 0) : s \in S_2 \cup S_3\}$, $(s', a') \in \{(s', 0) : s' \in S_0\}$ *we have* $m_{(s,a),(s',a')} = 0$.

Proof. We write the dot product above as a sum,

$$\boldsymbol{B}^{-1}_{(s,a)\cdot} \cdot \boldsymbol{N}_{\cdot(s',a')} = \sum_{k \in S} b_{(s,a),k} n_{k,(s',a')}.$$

By the previous proposition, we have that $b_{(s,0),k} = 0$ whenever $s = (q_1, q_2)$ and $k = (q_{1_k}, q_{2_k})$ such that $q_{1_k} + q_{2_k} < q_1 + q_2$ or $q_{1_k} < q_1$.

Similarly, recall that $n_{k,(s',a')} = p(k|s', a')$, so for every k such that $q_{1_k} + q_{2_k} \geq q_1' + q_2'$ and $q_{1_k} \geq q_1'$ we have for every $(s', a') \in \{(s', 0) : s' \in S_0\}$ that $n_{k,(s',a')} = 0$ because these are transitions from states such that $q_{1_k} + q_{2_k} > B$ to states such that $q_1' + q_2' < B$, which always have transition probability 0 under any action. ■

There has been some research exploring techniques to determine permanent basic or non-basic variables in an LP, that is, variables that must be present in every optimal solution or absent from every optimal solution, see Cheng [10, 11], Shi, Yu and Zhang [64], and more recently Paparrizos, Stephanides and Samaras [55]. Of particular interest to us is the work of Ye [74]. He proposed a technique for identifying permanent non-basic variables with or without degeneracy in the problem. Specifically, Corollary 3.3 in Ye [74] which we re-state here adapting it to maximization will be useful for us. It provides rules for when variables can be permanently eliminated from the problem.

Proposition 14 (Ye). *If there exists some state action combination (s, a) that has a corresponding column in $\boldsymbol{B}$, such that the variable $x(s, a) = 0$ and $m_{(s,a),(s',a')} > 0$, then do not eliminate the column for (s', a'); otherwise if*

$$\bar{c}(s', a') + \Delta \cdot \max\left(0, \max_{(s,a):x(s,a)>0} \left(\frac{m_{(s,a),(s',a')}}{x(s, a)}\right)\right) < 0$$

then the (s', a') column of $\boldsymbol{N}$ is non-basic for every optimal solution, where Δ is a non-negative upper bound on the optimal value.

In our case, we can take $\Delta = 0$ so the condition simplifies to: if for some state action combination (s, a) the variable $\bar{x}(s, a) = 0$ and $m_{(s,a),(s',a')} > 0$ do not eliminate column (s', a') otherwise if $\bar{c}(s', a') < 0$ then the (s', a') column of $\boldsymbol{N}$ is non-basic for every optimal solution.

Theorem 4. *For any optimal policy the decision at state (q_1, q_2) is to admit customer arrivals whenever $q_1 + q_2 < B$. Or, conversely, a deterministic policy that rejects customer arrivals at state (q_1, q_2) whenever $q_1 + q_2 < B$ cannot be optimal.*

Proof. It follows from Propositions 7 and 14 and Lemma 3 above, that the decision variables $x(s, a)$ such that $(s, a) \in \{(s, a) : s \in S_0, a = 0\}$ are permanently non-basic. Note that these are the only variables for which $\bar{c}(s, a)$ is always negative. Therefore the variables $x(s, a)$ such that $(s, a) \in \{(s, a) : s \in S_0, a = 1\}$ must be permanently basic, as the basis must have one variable for each state. ■

Define

$$\Pi' = \{\pi : \pi(s) = 1 \ \ \forall s \in S_0\}.$$

Then by the previous theorem it follows that if Π^* is the set of all optimal policies, $\Pi^* \subset \Pi' \subset \Pi$. Therefore, the set of states where a decision needs to be made can be reduced to $S_d = S_1 \cup S_2$. For every state in S_0 we can fix the action as admit (action 1). Similarly, for each state in S_3 the only available action is reject (action 0).

This result shows that no policy that admits in fewer states than the Prudent policy can be optimal. It also shows that the Prudent and Greedy policies are indeed opposite, in the sense that one is the most conservative policy that can be optimal and the other is the most greedy policy that can be optimal. Finally, note that any potentially optimal policy is a combination of the Prudent and Greedy policies. In the next section we provide further discussion on the structure of the optimal policy.

4.3.2 Structure of the Optimal Policy for Systems with $B = 2$

We begin by showing that, when $B = 2$, the optimal policy is of threshold type and provide the thresholds for each potentially optimal policy. Consider the policies for systems with $B = 2$ presented in Table 5. Note that the set of states where the decision is to admit is a monotonically increasing set as we move from the Prudent policy to $P1$ to $P2$ to the Greedy policy.

Table 5: Potentially Optimal Policies when $B = 2$

Policy	**Admit in States**	**Reject in States**
Prudent	(0,0);(0,1);(1,0)	(0,2);(1,1);(1,2);(2,0);(2,1);(2,2)
$P1$	(0,0);(0,1);(1,0);(1,1)	(0,2);(1,2);(2,0);(2,1);(2,2)
$P2$	(0,0);(0,1);(1,0);(1,1);(1,2)	(0,2);(2,0);(2,1); (2,2)
Greedy	(0,0);(0,1);(0,2);(1,0);(1,1);(1,2)	(2,0);(2,1); (2,2)

Proposition 15. *Consider a system with buffer capacity $B = 2$, then the Prudent, $P1$, $P2$ and Greedy policies, given in Table 5 are the only policies that can be optimal*

(up-to recurrent states). Furthermore, the optimal policy is determined by a series of monotonically decreasing thresholds on the ratio $\frac{c_2}{c_1}$. Hence as the ratio $\frac{c_2}{c_1}$ decreases, the set of states in which it is optimal to admit increases. And the order in which the states are added to the admission set does not dependent on the rates λ, μ_1, μ_2.

Proof. Start the Policy Iteration algorithm with the Prudent policy as the initial policy. Then perform the policy evaluation step to determine the gain and the bias. Next perform the policy improvement step. Here, we find that if

$$\frac{c_2}{c_1} \geq \left(1 + \frac{\mu_2^3(\lambda + 2\mu_1 + \mu_2)}{\lambda^2\left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right) + \lambda\mu_1\mu_2(\mu_1 + \mu_2) + \mu_1^2\mu_2^2}\right) \tag{10}$$

then the next decision rule is the same as before, and hence the Prudent policy is optimal. Otherwise, the Prudent policy is not optimal.

Now instead, start the policy iteration with policy $P1$. Then perform the policy evaluation step to determine the gain and the bias. Next perform the policy improvement step. Here, we find that if

$$\begin{aligned}\left(1 + \frac{\mu_2^3(\lambda + 2\mu_1 + \mu_2)}{\lambda^2\left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right) + \lambda\mu_1\mu_2(\mu_1 + \mu_2) + \mu_1^2\mu_2^2}\right) &\geq \frac{c_2}{c_1} \\ \geq \left(1 + \frac{\mu_2^3(\lambda + \mu_1 + \mu_2)\left(\mu_2(2\lambda + 3\mu_1) + (\lambda + \mu_1)^2 + \mu_2^2\right)}{\beta_2}\right)&\end{aligned} \tag{11}$$

where

$$\begin{aligned}\beta_2 =& 2\mu_2^4\left(\lambda^2 + \lambda\mu_1 + \mu_1^2\right) + \mu_2^3(\lambda + \mu_1)\left(3\lambda^2 + 4\lambda\mu_1 + 3\mu_1^2\right) + \mu_2^2\left(\lambda^2 + 3\lambda\mu_1 + \mu_1^2\right)^2 \\ &+ \lambda\mu_1\mu_2(\lambda + \mu_1)\left(\lambda^2 + 5\lambda\mu_1 + \mu_1^2\right) + \lambda^2\mu_1^2(\lambda + \mu_1)^2\end{aligned}$$

then the policy stays the same, and hence, $P1$ is the optimal policy. With some algebra one can show that the difference between the term on the left-hand side of condition (11) and the one on the right-hand side is positive.

Now, start the policy iteration algorithm with policy $P2$. Then perform the policy evaluation step to determine the gain and the bias. Next perform the policy improvement step. Here, we find that if

$$\left(1+\frac{\mu_2^3(\lambda+\mu_1+\mu_2)\left(\mu_2(2\lambda+3\mu_1)+(\lambda+\mu_1)^2+\mu_2^2\right)}{\beta_2}\right)\geq\frac{c_2}{c_1}$$
$$\geq\left(1+\frac{\mu_2^3\left(\mu_2^2(\lambda+\mu_1)+\mu_2(2\lambda+\mu_1)(\lambda+2\mu_1)+(\lambda+\mu_1)^3\right)}{\beta_1}\right) \tag{12}$$

then the policy stays the same, and hence $P2$ is the optimal policy. As in the previous case, one can verify that the difference between the left-hand side of condition (12) and the right-hand side is positive.

Finally, suppose you start the policy iteration with the Greedy policy. Then perform the policy evaluation step to determine the gain and the bias. Next perform the policy improvement step. Here, we find that if

$$\frac{c_2}{c_1}\leq\left(1+\frac{\mu_2^3\left(\mu_2^2(\lambda+\mu_1)+\mu_2(2\lambda+\mu_1)(\lambda+2\mu_1)+(\lambda+\mu_1)^3\right)}{\beta_1}\right) \tag{13}$$

then the Greedy policy is optimal. ■

Define p_s as the probability that a new arrival that is admitted when the system is in state s will be lost. A recurrent expression for calculating $1-p_s$ for each state is provided in Spicer and Ziedins [68]. Then, for the states where a decision must be made, the loss probabilities are:

$$p_{(1,1)}=\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^2 \;<\; p_{(1,2)}=\frac{\mu_1^2(\mu_1+2\mu_2)}{(\mu_1+\mu_2)^3} \;<\; p_{(0,2)}=\frac{\mu_1}{\mu_1+\mu_2}. \tag{14}$$

Note that the loss probabilities are not increasing in the total number of customers in the system, nor are they increasing in the number of customers at the first station. Similarly, we see that the decisions in an optimal policy are also non-monotone with respect to either of these parameters, when $P2$ is the optimal policy. From these probabilities and Proposition 15, we can immediately draw the following conclusion about the optimal policy, when $B=2$.

Corollary 3. *Suppose $B=2$, and a fixed set of parameters $\lambda,\mu_1,\mu_2,c_1,c_2$ and that π is an optimal policy, then:*

1. *For any pair of states $s, s' \in S_d$, which are recurrent under π if $p_s \geq p_{s'}$ we have $\pi(s) \leq \pi(s')$. That is, the optimal policy is monotone in p_s.*

2. *For any pair of states $s = (q_1, q_2)$ and $s' = (q_1, q_2')$, which are recurrent under π and $q_2' > q_2$, we have $\pi(s) \geq \pi(s')$. That is, for a fixed q_1 the optimal policy is monotone in q_2.*

3. *For any pair of states $s = (q_1, q_2)$ and $s' = (q_1', q_2')$, which are recurrent under π, $q_1 + q_2 = q_1' + q_2'$ and $q_2 < q_2' < B$, we have $\pi(s) \geq \pi(s')$. That is, for a fixed $q_1 + q_2$ the optimal policy is monotone with respect to q_2.*

4. *Consider now another pair of costs c_1', c_2', such that if $\frac{c_2'}{c_1'} \leq \frac{c_2}{c_1}$, for any optimal policy π' of the system with costs c_1', c_2', the following inequality holds: $\pi(s) \leq \pi'(s)$, for every s which is recurrent under π.*

Corollary 3 follows from Proposition 15. Specifically, Corollary 3.2, 3.3 hold because the Greedy, Prudent, $P1$ and $P2$ policies each have this structure, and are (up-to recurrent states) the only potentially optimal policies. On the other hand, Corollary 3.1 and 3.4 follow because the optimal policy when $B = 2$ is of threshold type, and the order of the thresholds in conditions (10), (11), (12) and (13) is the same for any set of parameters λ, μ_1, μ_2. Furthermore, the ordering of these thresholds is the same as the ordering of the loss probabilities for the states in S_d, given in (14), which is also independent of the parameters λ, μ_1, μ_2.

4.3.3 Counter-example and Discussion on Larger Buffers

Considering the above results, it seems intuitive to assume that Corollary 3 could hold for buffers of size $B \geq 3$. However, that is not always the case. There is no equivalent statement to Proposition 15 for the case where $B \geq 3$. The reason for this is that there does not exist a set of thresholds that determine the optimal policy and whose order is fixed regardless of the parameters. In this section, we show that Corollary 3 fails for larger systems using a counter-example.

Consider a system with the following parameters: $B = 3, \lambda = 23, \mu_1 = 74, \mu_2 = 80$ and costs $c_1 = 225, c_2 = 732$. Note that all costs and parameters are within an order of magnitude of each other, that is, this is not an extreme example. The optimal policy, π^*, for this case satisfies:

$$\pi^*(0,0) = \pi^*(0,1) = \pi^*(0,2) = \pi^*(1,0) = \pi^*(1,1) = \pi^*(1,2) = \pi^*(2,0) = \pi^*(2,1) = 1,$$

$$\pi^*(0,3) = \pi^*(1,3) = \pi^*(2,2) = \pi^*(2,3) = \pi^*(3,0) = \pi^*(3,1) = \pi^*(3,2) = \pi^*(3,3) = 0.$$

In this case, we have that the loss probabilities of states $(1,2)$ and $(2,2)$ are

$$p_{(1,2)} = 0.230 \; > \; 0.226 = p_{(2,2)}.$$

So, while customers arriving in state $(1,2)$ have a greater probability of being lost than those arriving in state $(2,2)$, it is optimal to admit customers in state $(1,2)$ but not in state $(2,2)$. In other words, the optimal policy is not monotone with respect to the loss probabilities, so Corollary 3.1 does not hold for $B \geq 3$.

For cases where the buffers $B \geq 3$, there is no fixed ordering of the states based on the loss probabilities (unlike the $B = 2$ case). Furthermore, there is no monotonically increasing admission set, as in the $B = 2$ case. So, even though for a given set of parameters λ, μ_1, μ_2 the optimal policy may be threshold type, the order of the thresholds can vary with the parameters.

On the other hand, numerical experiments suggest that Corollary 3.2-3.4 also seem to hold for larger values of B. Specifically, for each value of $B = 3, 4, 5, 6$ we randomly generated 300,000 instances, with λ, μ_1, μ_2 and c_1 drawn from a uniform distribution between 0 and 1000, and c_2 drawn for a uniform distribution between c_1 and 1000. For each instance we solved the MDP using the Policy Iteration algorithm (see Puterman [58], Section 8.6.1). For every instance we tested, the optimal policy had the structure described in Corollary 3.2 and 3.3. Furthermore, we obtained c_2' by drawing a uniformly distributed factor between 1 and 10 and multiplying the original c_2 by this factor. We solved a new MDP substituting c_2 for c_2' using the policy iteration algorithm, then compared the optimal policy of each pair. Corollary 3.4 also held for every instance we tested.

Corollary 3.2 is intuitive, as a customer arriving in a state that is busier at the second station and the same at the first station is more likely to suffer a loss. So, if the gatekeeper rejects arrivals at a given state, he should also reject arrivals in states that have the same number of customers at the first station and are busier at the second station. Corollary 3.3 is also intuitive, because a customer that observes $q_1 + q_2$ customers upon arrival is more likely to experience a loss if there are more customers at the second station. So, if the gatekeeper rejects arrivals at a given state, he should also reject arrivals in states that have the same total number of customers and are busier at the second station.

Finally, the fact that Corollary 3.4 seems to hold would suggest that, for any given instance of the problem, there exists an optimal policy which is threshold type. Therefore, a policy that has a threshold structure that closely approximates the exact thresholds for each state, may deliver near-optimal performance. On the other hand, if the range of values where the Prudent and Greedy policies are not optimal is narrow, then a strategy that uses only these policies may also be near-optimal. In the following section we use these two insights and the results for the Prudent and Greedy policies to propose heuristic policies which are easy to implement, and then we evaluate the long-run average cost performance of these policies using numerical experiments.

4.4 Heuristics

So far, we have shown that if $\frac{c_2}{c_1} \geq c_*(B)$, then the Prudent policy is optimal and if $\frac{c_2}{c_1} \leq 1$ then the Greedy policy is optimal. The only case left where the system administrator needs to make a decision is when $c_*(B) < \frac{c_2}{c_1} < 1$. Nevertheless, in Section 4.3, we demonstrated that the optimal policy may have a complicated form. For very small systems, it may be possible to solve the MDP to optimality using the classical MDP techniques such as policy iteration. However, for large systems this may not be practical. Furthermore, it is desirable to provide the system administrator with a simple, easy-to-implement admission control policy to achieve optimal (or near-optimal) performance in all instances. In this section, we construct two heuristic policies, based on insights from the aforementioned results. We perform numerical tests which suggest some of these heuristics can provide near-optimal performance in most instances.

We consider the following four policies:

- **P**: The Prudent policy: we include this policy for benchmarking purposes. This policy is intuitive because it guarantees that customers are never lost.

- **G**: The Greedy policy: we also include this policy for benchmarking purposes. It is equivalent to not exercising any admission control.

- **PvsG**: This heuristic policy leverages the fact that we have an exact characterization of the long-run average cost under the Prudent policy and a matrix analytic solution for the long-run average cost of the Greedy policy. This heuristic evaluates the Prudent policy using Proposition 6 and the Greedy policy using Proposition 10, and then chooses the best of these two. It is computationally easier to evaluate these policies, than to run an algorithm to find the optimal policy. Evaluating the Greedy policy still requires solving a linear system of equations of size $3(B + 1)$.

- **LossProb**: This is a two-step heuristic. First, it determines whether the Prudent policy is optimal, using the threshold $c_*(B)$ from Theorem 3. If the Prudent policy is optimal it uses the Prudent policy. Otherwise, for each state s the heuristic calculates the probability p_s that a new arrival is lost. For each state $s \in S_d$ set:

  $$\pi(s) = \mathbf{1}_{\{p_s \le c_1/c_2\}}$$

  This heuristic uses the conjecture that the optimal action is monotone with respect to $\frac{c_2}{c_1}$ for each state, and uses the loss probabilities p_s as proxies for the actual thresholds (which are unknown). Although in Section 4.3 we provide counter-examples that show that this ordering of states is not necessarily the same as the ordering in the optimal policy, we have observed in numerical tests that this order matches the optimal policy ordering in up to 95% of cases (for $B = 3$).

The two heuristic policies offer very different strategies. The PvsG heuristic leverages the fact that we have efficient means to price the Prudent and Greedy policies and never attempts to calculate the optimal policy. On the other hand, the LossProb heuristic tries to directly approximate the optimal policy, based on the observation that it appears to be of threshold type for each state, and does not rely exclusively on the Prudent and Greedy policies. Computationally, The LossProb heuristic is more efficient than the PvsG heuristic, as the loss probabilities can be calculated recurrently, while evaluating the greedy policy requires solving a system of $3(B+1)$ linear equations.

In order to test the proposed heuristics we use the same 1000 randomly generated data sets, we described in Section 4.2.2. We call each of these data sets an instance. Recall that for each instance we draw λ, μ_1 and μ_2 independently from a continuous uniform distribution in (0,100). Similarly we generate $c_1, c_2 \sim U(0, 1000)$ and discard the values $c_1 \geq c_2$ (as otherwise the Greedy policy is optimal as stated in Proposition 12).

In order to compare which heuristic policies work better for systems of different sizes we split our tests into three different categories:

- **Small systems:** $B \in \{1, 2, 3\}$.
- **Medium systems:** $B \in \{4, 5, 6, 7, 8, 9, 10\}$.
- **Large systems:** $B \in \{15, 20, 25, 30\}$.

We define a system as each value of B taken from one of the three sets above. For each system, we test all 1000 instances and for each one we calculate the long-run-average cost of using each heuristic policy, as well as the cost of the optimal policy, which we find using the policy iteration algorithm. In what follows we summarize our results and highlight some insights that we draw from them.

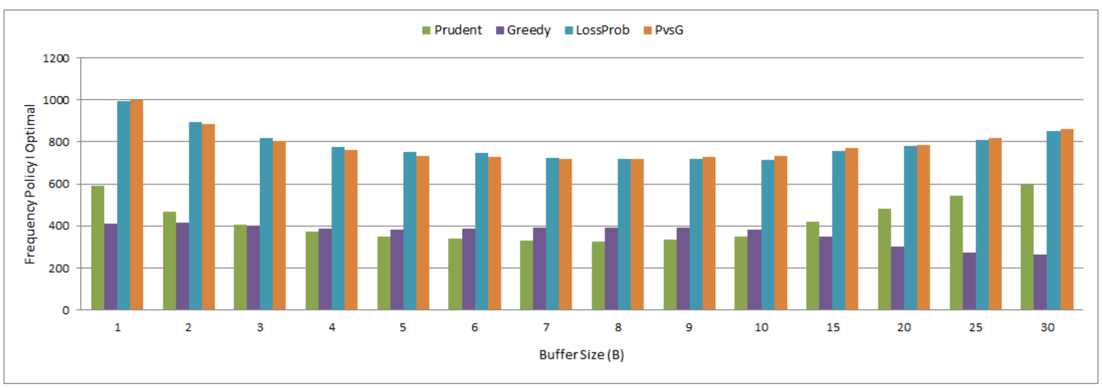

Figure 9: Number of instances that each heuristic is optimal.

Figure 9 shows the number of instances that each of the heuristics chooses the optimal policy. We see that the Prudent policy is optimal in about 40% to 60% of the test instances. This is expected as we are not limiting our instances to those where $\frac{c_2}{c_1} > c_*(B)$, so it is possible that we often have situations where the Prudent policy is optimal. We also note that the Prudent policy is optimal more frequently when the system is small or large, but less frequently for medium systems. We expect the

Prudent policy to be optimal often for small systems as the set of potentially optimal policies is quite small. For the medium systems there are more potentially optimal policies, so the Prudent policy is optimal less often. Finally, for large systems, the Prudent policy is optimal more often than for medium systems. This is consistent with the results of Section 4.1.2. As B increases, the gain of the Prudent policy also increases, and in particular for cases where $\lambda < \min(\mu_1, \mu_2)$, it approaches the optimal gain as $B \to \infty$. So it is expected that as B increases, the Prudent policy should be optimal more often.

On the other hand, the Greedy policy is optimal about 30% to 40% of instances for most system sizes. This again is expected as the condition $c_1 > c_2$ is sufficient but not necessary for optimality. We observe that the number of instances where the Greedy policy is optimal reduces slowly as system size increases. This is also expected, as we pointed out in Section 4.2.2 unless the stability condition holds it is possible that the Greedy policy performs worse as the size of the buffer increases.

Both the LossProb and PvsG heuristic policies are optimal more than 80% of instances for small systems and about 70% of instances for medium systems and between 70% and 80% of instances for large systems. These results provide encouraging evidence that these heuristics may be a good alternative to finding the optimal policy. The results suggest that in medium systems there is more to be gained from the additional work of finding the optimal policy, than in small or large systems. Also, neither heuristic dominates the other in terms of how frequently each achieves optimality. However, as system size increases, it appears that the PvsG heuristic performs slightly better than the LossProb heuristic.

Now we analyze the results in terms of average performance of each policy relative to the optimal policy. We compare the cost (not the gain), as this is a more intuitive, but equivalent measure. We begin our analysis with small systems. Table 6 shows the percentage of excess cost over the optimal long-run-average cost for each of the

Table 6: Average percentage additional cost over the optimal cost in small systems.

B	Prudent	Greedy	LossProb	PvsG
1	2.719 ± 0.301%	44.356 ± 12.908%	0.000 ± 3.10E-04%	0.000 ± 1.06E-06%
2	5.382 ± 0.590%	56.609 ± 18.296%	0.139 ± 4.19E-02%	0.101 ± 3.09E-02%
3	7.020 ± 0.801%	65.014 ± 22.418%	0.356 ± 9.15E-02%	0.220 ± 5.71E-02%

heuristics for $B = 1, 2, 3$ (averaged over all 1000 test instances) and the 95th percentile confidence intervals. It is clear that both heuristics are on the average within 1% of optimal. The Prudent policy performs one order of magnitude worse, with the average for each case being less than 10% over optimality. The Greedy policy performs the worst, at up to 65% above optimality. These results highlight the importance of choosing the correct policy as naively choosing the Greedy policy can result in enormous excess cost. Table 6 also shows that both heuristics yield near optimality for small systems. Note that the difference in long-run-average cost between the LossProb and PvsG heuristics, for small systems, is not statistically significant. On the other hand, the differences between any other pair of policies are statistically significant. We conclude that both heuristics outperform the Prudent policy, which outperforms the Greedy policy.

Table 7: Average percentage additional cost over the optimal cost in medium systems.

B	Prudent	Greedy	LossProb	PvsG
4	8.202 ± 0.995%	66.003 ± 25.187%	0.542 ± 0.128%	0.273 ± 0.067%
5	8.671 ± 1.144%	70.417 ± 27.952%	0.709 ± 0.165%	0.326 ± 0.085%
6	8.098 ± 1.172%	75.099 ± 30.647%	0.970 ± 0.229%	0.356 ± 0.092%
7	8.037 ± 1.254%	78.728 ± 32.753%	1.230 ± 0.288%	0.374 ± 0.098%
8	6.936 ± 1.131%	81.995 ± 34.742%	1.301 ± 0.318%	0.404 ± 0.110%
9	6.301 ± 1.100%	85.183 ± 36.505%	1.404 ± 0.351%	0.423 ± 0.123%
10	6.103 ± 1.111%	87.233 ± 37.579%	1.548 ± 0.394%	0.435 ± 0.135%

Next we consider medium systems. Table 7 shows the percentage of excess cost over the optimal long-run average cost for $B = 4, \ldots, 10$ (averaged over all 1000 test instances) and the corresponding 95th percentile confidence intervals. The data shows that both heuristics outperform the Greedy and the Prudent policies. Furthermore, it is apparent that there is a greater penalty for using the Greedy policy when it is not optimal than using the Prudent policy when it is not optimal. Table 7 shows both heuristics are on the average within 2% of the optimal cost. In this case, the difference in average excess cost for any pair of policies is statistically significant. So, we can say that the PvsG heuristic has the best performance, followed by the LossProb heuristic, the Prudent policy and the Greedy policy. Recall however, that the PvsG heuristic requires more computation than the LossProb heuristic, so if this is an important factor for the system administrator, the LossProb heuristic may be a better alternative for medium systems.

Table 8: Average percentage additional cost over the optimal cost for large systems.

B	Prudent	Greedy	LossProb	PvsG
15	$3.639 \pm 0.863\%$	$97.097 \pm 42.561\%$	$1.534 \pm 0.504\%$	$0.254 \pm 0.097\%$
20	$2.476 \pm 0.713\%$	$101.383 \pm 44.877\%$	$1.556 \pm 0.596\%$	$0.151 \pm 0.068\%$
25	$2.080 \pm 0.685\%$	$103.474 \pm 45.788\%$	$1.389 \pm 0.640\%$	$0.116 \pm 0.058\%$
30	$1.406 \pm 0.565\%$	$105.264 \pm 46.868\%$	$1.314 \pm 0.681\%$	$0.068 \pm 0.035\%$

Finally, we consider large systems. Table 8 compares the excess long-run-average cost over the optimal cost for $B = 15, 20, 25, 30$. Again, there is a greater cost for using the Greedy policy when it is not optimal. In fact, from Tables 6, 7 and 8, we can conclude that the performance of the Greedy policy gets worse as the buffer size increases. This is consistent with our findings that the gain under the Greedy policy is not monotone decreasing in most cases. Here, we observe that for very large systems the Prudent policy performs well, in some cases as well as the heuristics. Note that the difference in average excess cost between the Prudent policy and the LossProb heuristic is not statistically significant. The difference in average excess cost between any other pair of policies is statistically significant. So, we can state that the PvsG heuristic delivers the best performance. Next in performance are the LossProb heuristic and the Prudent policy. The Greedy policy performs the worst on the average. Hence, for large systems it is best to use the PvsG heuristic. If computational capacity is very limited the best alternative is to use either the Prudent policy or the LossProb heuristic. We should note that although the performance of the Prudent policy and the LossProb heuristic are similar on the average, Figure 9 shows that the LossProb heuristic chooses the actual optimal policy more often than the Prudent policy. Hence, if the administrator is also interested in increasing the likelihood of choosing the optimal policy, the LossProb heuristic is a better choice than the Prudent policy.

Next, we focus on the worst-case performance of each of the policies tested. Table 9 shows the percentage of excess cost over the optimal long-run average cost for each of the heuristics in the worst case of the 1000 test instances for each value of B that we tested. In all cases we observe that the Greedy policy always has the highest worst-case performance in excess cost. The Prudent policy performs approximately one order of magnitude better in terms of worst-case excess cost. For small and medium systems the PvsG heuristic is an order of magnitude below the Prudent policy in worst-case performance, while the LossProb heuristic performs twice as badly, but better than the Prudent policy. For large systems the PvsG heuristic clearly performs better than the LossProb heuristic, which performs similarly to the Prudent policy. Overall, the excess cost of the PvsG heuristic is never more than 30% over the optimal cost. The LossProb heuristic and the Prudent policy are always within a factor of 2.5 of the optimal cost. The Greedy policy is between a factor of 30 and 100 times the optimal cost in the worst case.

We conclude that if the system administrator must choose a naive policy, the Prudent policy is a better choice than the Greedy policy. However, these results

Table 9: Worst Case percentage additional cost over the optimal cost.

B	Prudent	Greedy	LossProb	PvsG
1	28.8%	3462.6%	0.1%	0.0%
2	55.7%	4884.1%	7.4%	5.2%
3	73.6%	5871.7%	17.5%	10.4%
4	91.0%	6637.3%	21.3%	10.1%
5	103.4%	7246.8%	24.8%	14.6%
6	111.3%	7724.8%	29.8%	13.3%
7	123.3%	8097.9%	33.3%	14.4%
8	106.2%	8391.3%	37.0%	18.0%
9	118.8%	8621.2%	40.9%	18.5%
10	135.0%	8803.9%	45.0%	23.8%
15	105.8%	9304.5%	65.5%	15.5%
20	105.1%	9496.5%	85.2%	11.8%
25	102.2%	9584.6%	103.6%	10.9%
30	90.4%	9620.5%	120.3%	5.9%

show that always using either the Prudent or the Greedy policy is too naive and may result in large excess costs, regardless of system size. Furthermore, the results of this section show that just using the findings about the Greedy and Prudent policies provided in this chapter, possibly in conjunction with the loss probabilities, the system administrator can achieve near-optimal long-run average cost performance.

4.5 Conclusions

In this chapter we consider a similar problem to the one we explored in Chapter 3, but now allow for identical, finite, arbitrary buffers at both stations. First, we focus on the Prudent policy, which only admits an arrival if the loss probability is zero. For the Prudent policy we provide a closed form expression for the long-run average reward, as well as necessary and sufficient conditions for optimality. We also show that the long-run average cost under this policy is monotone, and calculate its limit as $B \to \infty$. Then, we turn to the Greedy policy, which admits an arrival as long as there is space at the first station. For the Greedy policy we give a matrix-analytic solution for the long-run average cost and a sufficient condition for optimality.

Next, we explore the structural properties of the optimal policy. First, we identify a subset of states where it is always optimal to admit incoming arrivals. This shows, that no policy admits customers in fewer states than the Prudent policy. We show that the optimal policy is not monotone with respect to the total number of customers in the system or the length of the queue at the first station, nor with respect to the loss probabilities of an arriving customer. We also provide a complete characterization of the optimal policy for the special case where the buffer at each station is two, and we show that for larger buffers the optimal policy can have a complicated structure.

Lastly, using the results about the Prudent and Greedy policies, together with insights on the structure of the optimal policy, we design two heuristic admission control policies. Numerical experiments show that always using either the Prudent or Greedy policy is too naive and may result in average excess cost of up to 100% and 10000%, respectively. Furthermore, the results indicate that using the results provided here to evaluate the performance of the Prudent and Greedy policies and picking the one with the smaller long-run average cost value yields near-optimal performance. Alternatively, using a heuristic based on the loss probabilities of the arriving customer can yield near-optimal performance for systems of certain sizes, requiring fewer computations.

CHAPTER V

DYNAMIC CONTROL OF COMPLEX AUTHENTICATION SYSTEMS

In this chapter, we consider a different admission and routing control problem. We study a centralized system where requests for authentication arrive from different users. The system has multiple authentication methods available and a controller must decide how to assign a method to each request. We consider three different performance measures: usability, operating cost, and security. We model each authentication method as a multi-server queue and model the trade-offs between these performance measures using a cost-based approach and a constraints-based approach. For each approach we construct a Markov Decision Process (MDP) in order to derive structural and computational results on the optimal admission and routing policy. We also provide a numerical example to illustrate the trade-offs between the three performance metrics, and show how to use our models to build an efficient frontier.

This chapter is organized as follows: Section 5.1 presents background information and a more detailed description of the problem. Section 5.2 gives an overview of the literature about this specific control problem. Section 5.3 presents a detailed mathematical model of the system. In Section 5.4, we present a cost-based approach to the problem. Using a MDP model, we find the structure of the optimal policy under certain assumptions about the system characteristics. In Section 5.5, we introduce a constraint-based approach using a constrained MDP model. Here we also find the structure of the optimal policy under specific assumptions. Section 5.6 offers a numerical example for illustration purposes. Finally, Section 5.7 concludes the chapter. A summarized version of the main results in this chapter can be found in

Silva, Zhang and Ayhan [67].

5.1 Background

As mobile technology becomes ubiquitous, people are using their devices for more tasks than ever. Many of these tasks require authentication, such as banking, making purchases and sending secure communications. At the same time, modern mobile devices are equipped with more hardware and sensors as standard features, including multi-touch screens, biometric sensors, GPS and accelerometers (Bao *et al.* [4]). In consequence, we are experiencing the simultaneous growth of demand for authentication and a proliferation of methods to perform it. Hence, there has been a significant increase in the challenge of managing the hardware and software available for authentication.

Authentication is a secondary task, and therefore latency in this process can cause user dissatisfaction. We refer to the system's *usability* as a general measure of the user experience, which includes their dissatisfaction with latency. Extended latency in secondary tasks has been linked with adverse effects on productivity and short term memory (Trafton, Altmann and Brock [70]). Therefore, it is desirable to implement authentication schemes that minimize latency. On the other hand, more secure methods of authentication often cause greater latency, and thus lower usability. So, there is a trade-off between security and usability. It is also desirable for the system administrator to use an authentication scheme that minimizes the system's overall operating cost, but in practice the cheaper authentication methods may have lower usability and/or be less secure. Thus, there is a three-way trade-off between usability, operating cost and security.

In order to adapt to changing conditions, the authentication system controller must consider the state of the system when making decisions. For example, when the system is experiencing high congestion, it may be better to incur additional cost in order to reduce overall latency until the system reaches a less congested state. The system must also consider the characteristics of the user and those of the request. For instance, a request to transfer funds from a bank account may be subject to more stringent security than one for checking its remaining balance. Similarly, requests from premium customers may have higher usability expectations for the system to meet than those from regular customers. Our objective is to develop a methodology to manage complex authentication schemes. Specifically, we develop a stochastic dynamic control approach to assigning different authentication methods to incoming tasks, taking into account all three performance metrics, namely, usability, operating cost, and security. In most cases, we focus on balancing cost and security, but we also track how the methods we propose impact usability. In a few special cases we propose methods to balance all three objectives simultaneously.

More specifically, we consider a centralized authentication system where requests with distinct characteristics arrive from different users. The system has multiple authentication methods available and a controller must decide how to assign an authentication method to each request, based on the current state of the system and the probability that the request comes from an impostor. In many applications each request is associated with information about the user's history, location, device, network, the request's nature, priority, urgency, and various other intrinsic and contextual attributes. However, we assume that there is a pre-processing step in place, which uses those attributes to estimate each request's probability of being legitimate based on such information, these probabilities are the inputs to our model. The controller makes authentication assignment decisions using only the estimated probabilities. The pre-processing step can be viewed as a module based on appropriate statistical or machine learning algorithms. The development of such algorithms is an interesting research topic by itself, and outside the scope of this thesis.

The dynamics of the authentication system are represented by queueing models with and without delay, depending on the amount of capacity available for each method. We use results from queueing theory, as well as those on constrained and unconstrained MDPs to develop a strategy for dynamically assigning incoming requests to authentication methods that balance operational cost, latency and security.

5.2 Related Work

The literature available on this problem is limited. The existing literature focuses on the security aspect. Historically, authentication systems used a single form of authentication and treated all requests equally in all situations. Using available information about users or authentication requests, was first considered in Denning and MacDoran [16], where they propose confirming physical location as part of the authentication process. The need to make authentication decisions based on the state of the system was pointed out in Cheng [12], where authentication decisions are based on a measure of the level of risk to the system. In Clark *et al.* [13], they add the feature of parameter uncertainty and time-sensitivity to the risk-based authentication model. However, none of these models considers latency. The importance of the effects of latency was particularly highlighted in modern mobile computing settings due to the fact that the primary task via interaction with mobile devices can be so brief that it is dominated by the time to authenticate, see for example Trafton *et al.* [70] and Nagata [52].

A recent short paper by Kovad and Zhang [37], suggests preliminary ideas on queueing theory-based methods for modeling complex authentication systems. The present chapter incorporates some of those ideas in the detailed approach developed here. We draw on various sources from the literature regarding admission and routing control into networks of parallel queueing systems detailed in Chapter 2. To the best of our knowledge, this is the first application of queueing theory and stochastic control to dynamically control a complex authentication system.

5.3 *Mathematical Model*

We consider an authentication system where requests for authentication arrive from different users. A single central controller must decide what authentication challenge to issue to each incoming request. Based on their characteristics, a pre-processing procedure has classified user requests into I classes, where each class consists of user requests with the same likelihood of coming from an impostor. We assume that the classification procedure is given, and that the controller only sees each request's class. Specifically, each request from class i has a fixed probability p_i of coming from an impostor, which is known to the controller as well. So the probability that a request for authentication of class i comes from a true user is $1-p_i$. We assume requests from each class arrive to the system following independent Poisson processes with rate λ_i for each $i = 1, 2, ..., I$.

The system controller has full knowledge of the system state and can observe which class each incoming request belongs to. It must decide how to assign these incoming requests among J different available authentication methods. These methods may include a password challenge, verification of biometric data, or any other data that can be provided by the user (e.g., voice, image, video). Each authentication method j has independent and identically distributed (i.i.d.) service times, with mean $1/\mu_j$. These times are independent of each other and of the arrival processes. A graphical representation of the system is provided in Figure 10.

We measure the level of security using two conditional probabilities: the probability of not authorizing a request given that it does not come from an impostor, also known as Type-I error probability; and the probability of authorizing a request given that it comes from an impostor, also called the Type-II error probability. Specifically, each method j's Type-I error probability is denoted by α_j and Type-II error

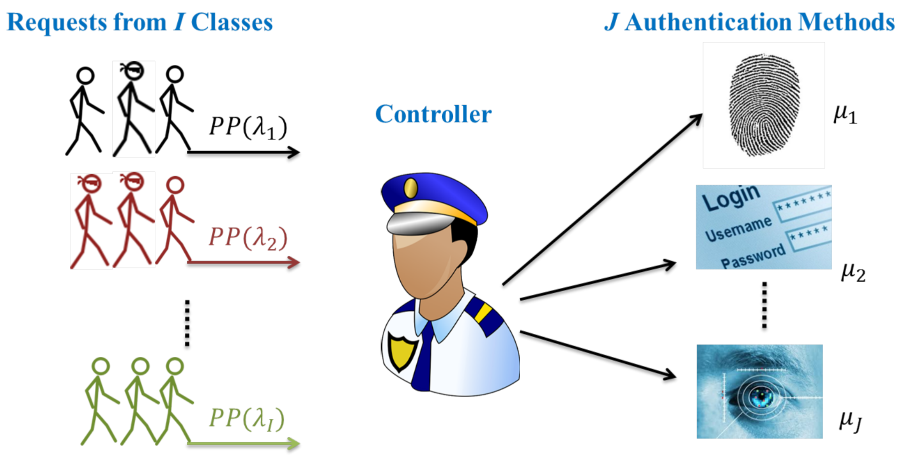

Figure 10: The authentication method assignment problem.

probability by β_j. We measure operating cost by charging a fixed cost c_j each time authentication method j is used. To measure usability, we define a latency cost which is proportional to the time that a request spends in the system (either waiting or in service). Let h_{ij} be the latency cost per unit-time for type i requests assigned to method j.

Our goal is to determine how to assign each incoming request to an appropriate authentication method. Specifically, we define an assignment policy as a function π whose input is the current state of the system and the class index of the incoming request, and whose output is either a single action to be taken, i.e., a deterministic policy, or a probability distribution over all the available actions, namely, a randomized policy. We aim to find an assignment policy that optimizes the long-run average performance of the system in terms of its operating cost, security, and usability.

We develop two approaches, which depend on the parameters available to the controller. In Section 5.4 we present a cost-based approach, which considers penalty costs each time an error is committed. In Section 5.5 we introduce a constraint-based approach which assumes that there is a bound on the percentage of requests that are allowed to result in errors. For each approach, we present analytical results for policies that optimize the operating cost and security, but not usability. However, we also prescribe explicit formulas for assessing the usability measure, even if it is not considered in the optimization process. Additionally, in the cases where all authentication methods in the system have infinite processing capacity, we provide methods to find policies that achieve optimal performance in terms of all three metrics.

5.4 Cost Based Approach

In order to capture the trade-off between operating cost and security we assign a penalty cost for committing errors. Specifically, we charge a cost C_I for committing Type-I error and C_{II} for committing Type-II error. These costs are the same for every authentication method and request class. The expected Type-I and Type-II error costs will be determined by the total probability that a request of class i that is assigned to method j results in an error. The Type-I error probability is given by $\alpha_{ij} := (1 - p_i)\alpha_j$ and the Type-II error probability will be given by $\beta_{ij} := p_i\beta_j$. Then the expected total cost of assigning a class i request to method j (including operational cost and expected security cost) is given by:

$$c_{ij} := c_j + \alpha_{ij}C_I + \beta_{ij}C_{II}.$$

In order to determine an optimal policy it is necessary to make assumptions about the capacity available for each authentication method. We solve the problem under two capacity assumptions. First, we assume that each method has infinite processing capacity. This is in line with systems used in practice, where authentication is handled by data centers with very large capacity, and authentication methods such as password verification or security questions require very few resources. Next, we assume that one method in particular has finite capacity, while all others have infinite capacity. Again, in practice one (or a few methods) may require some resource with limited availability, such as a confirmation phone call to the user by a human agent.

5.4.1 Infinite Capacity for All Methods

If the capacity available to process requests at each method is infinite, then we can model each authentication method as an $M/G/\infty$ queue. That is, a queueing system where arrivals follow a Poisson process, there are infinitely many servers and service times are i.i.d. with a general probability distribution.

Note that, because each method has an infinite number of servers, the decision to admit an arriving customer to a certain queue will not affect the capacity available to subsequent customers. So, the optimal strategy for the controller is static, that is, it does not depend on the current number of customers in the system, or on past decisions. Hence, the optimal policy can be found by evaluating c_{ij} for each method j, for each class i and assigning the method $j^*(i)$ to class i where

$$j^*(i) = \min_j \{c_{ij}\}.$$

The optimal policy can be stated as: whenever an incoming request is of class i, assign it to method $j^*(i)$.

In this system, because there are infinitely many servers, the waiting time is zero for every arrival. Hence, the expected latency cost incurred by a request of class i that is assigned to method j will be $\frac{h_{ij}}{\mu_j}$. Furthermore, in this case we can determine a policy which balances the three way trade-off between usability, operating cost and security by defining

$$j^*(i) = \min_j \left\{ c_{ij} + \frac{h_{ij}}{\mu_j} \right\}.$$

In this case the optimal policy is stated as before: whenever an incoming request is of class i, assign it to method $j^*(i)$.

5.4.2 One Method with Finite Capacity

In this section, we consider the case of two methods, where Method 1 has a finite capacity of M with m available servers whereas Method 2 has infinite servers. For

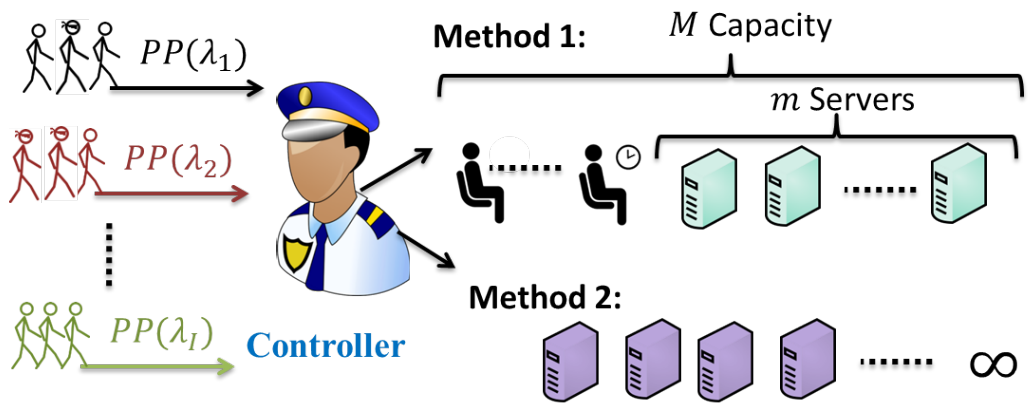

Figure 11: The finite capacity authentication method assignment problem.

simplicity of exposition, we assume there is only one method with infinite capacity. However, the results in this section hold for an arbitrary finite number of authentication methods each of which has infinitely many servers. It is only necessary to group the best infinite capacity method for each request class $j^*(i)$ into a virtual method, and call that Method 2. We assume the server for Method 1 has an exponentially distributed service time with service rate μ_1. Method 2 has i.i.d. generally distributed service times with rate μ_2. Figure 11 represents this scenario graphically. For tractability, in the remainder of this section we focus on finding policies that balance operating costs, with security costs, not taking into account latency costs. However, at the end of the section we provide an expression to calculate the latency cost of a given policy.

Let $\mu_{1,k}$ be the service rate for Method 1 when there are k customers there. That is $\mu_{1,k} := \min(k, m)\mu_1$. And let $\Lambda := \sum_{i=1}^{I} \lambda_i$ be the total arrival rate.

The problem is a continuous-time stochastic process, but we can model it as a discrete-time MDP using uniformization, as in Lippman [48]. We follow a similar procedure as in previous chapters. We observe the system only at discrete times immediately after certain events. Note that since Method 2 has infinite capacity, the number of customers in service at that method will not affect the decision to assign a new customer to either method, therefore we can ignore the job completion events at

Method 2. Furthermore, we assume that all the Method 1 servers work at all times. If they are serving an actual customer, we call that a *real* service; otherwise we call it a *fictitious* service. Then, since inter-arrival times for all customer types and service times follow an exponential distribution, we can use the uniformization procedure and observe the process immediately after each event of the following types: (a) a new customer arrival, and (b) a (real or fictitious) service completion at Method 1. Then the time between events follows an exponential distribution with parameter, $(\mu_{1,m} + \Lambda)$ which (without loss of generality) we assume is equal to 1.

Now we proceed to define the MDP. We use the sub-index n to define the n-th event. Let $X_n \in \{0, 1, 2..., M\}$ be the number of customers at Method 1 at the time of the n-th event. Let $Y_n \in \{0, 1, 2, ..., I\}$ define the event just observed in the system, where $Y_n = 0$ means the event was a (real or fictitious) service completion, and $Y_n = i \geq 1$ means the last event was a type i arrival. Then we define the state of the MDP at instant n as the tuple (X_n, Y_n), with a state space $\mathcal{S}$. The available actions at each state s are:

$$\mathcal{A}_s = \begin{cases} \{0\} & s = (k, 0), k \in \{0, 1, 2, ..., M\}, \\ \{1, 2\} & s = (k, i), k \in \{1, ..., M\}, i \in \{1, ..., I\}, \\ \{2\} & s = (k, i), k = M, i \in \{1, ..., I\}. \end{cases}$$

Action 0 represents doing nothing, and actions 1 and 2 stand for assigning the request that just arrived to Method 1 or 2, respectively. Recall the assumption that $\mu_{1,m} + \Lambda = 1$. Let $p(s'|s, a)$ denote the transition probabilities from state s to state s' when action a is taken in state s. First consider the probability that the next event is an arrival:

$$p((k, l)|(k, 0), 0) = \lambda_l \quad \forall l \in \{1, ..., I\}, k = 0, 1, ..., M$$

$$p((k + 1, l)|(k, i), 1) = \lambda_l \quad \forall i, l \in \{1, ..., I\}, k = 0, 1, ..., M - 1$$

$$p((k, l)|(k, i), 2) = \lambda_l \quad \forall i, l \in \{1, ..., I\}, k = 0, 1, ..., M.$$

Now consider the probability that the next event is a real service completion (this event is only possible if $k \geq 1$):

$$
\begin{aligned}
p((k-1,0)|(k,0),0) &= \mu_{1,k} \quad k = 1, ..., M \\
p((k,0)|(k,i),1) &= \mu_{1,k} \quad \forall i \in \{1, ..., I\}, k = 1, ..., M \\
p((k-1,0)|(k,i),2) &= \mu_{1,k} \quad \forall i \in \{1, ..., I\}, k = 0, 1, ..., M.
\end{aligned}
$$

Finally, consider when the event is a fictitious service completion. Note that this event is only possible when $k \leq m-1$:

$$
\begin{aligned}
p((k,0)|(k,0),0) &= 1 - \Lambda - \mu_{1,k} \quad \forall k = 0, 1, ..., m-1 \\
p((k+1,0)|(k,i),1) &= 1 - \Lambda - \mu_{1,k+1} \quad \forall i \in \{1, ..., I\}, k = 0, 1, ..., m-1 \\
p((k,0)|(k,i),2) &= 1 - \Lambda - \mu_{1,k} \quad \forall i \in \{1, ..., I\}, k = 0, 1, ..., m-1.
\end{aligned}
$$

All other transition probabilities $p(s'|s,a) = 0$.

To complete the definition of the MDP, we define the immediate reward $r(s,a)$ of taking action a at each state s. These are:

$$
\begin{aligned}
r((k,i),0) &= 0 \quad \forall i \in \{1, ..., I\}, 0 \leq k \leq M, \\
r((k,i),1) &= -c_{i1} \quad \forall i \in \{1, ..., I\}, 0 \leq k \leq M-1, \\
r((k,i),2) &= -c_{i2} \quad \forall i \in \{1, ..., I\}, 0 \leq k \leq M.
\end{aligned}
$$

Note that for any stationary deterministic or randomized policy, the resulting model is a discrete-time Markov chain. Under any stationary policy state $(0,0)$ is reachable from any state if sufficiently many transitions occur without an arrival. Therefore, this state is recurrent under every stationary policy and every other state is either in the same recurrent class or transient. Hence, the model is unichain, that is, under any policy, the resulting Markov chain has a single recurrent class. Note that state $(0,0)$ is also aperiodic, since a fictitious service triggers a transition to itself. Because the model is unichain and aperiodic, it has finite state and action spaces and bounded rewards, this implies that there exists an optimal policy that is stationary and deterministic (see Theorem 8.4.2 in Puterman [58]). So, we consider only these policies. We define a stationary deterministic policy π as a function that takes the state of the system $s \in \mathcal{S}$ and returns an action $a \in A_s$.

The objective is to find a policy π^* that maximizes the long run average reward. Note that all rewards are non-positive, because they represent costs. Define $\mathbb{E}^\pi$ as the expectation under policy π, then the long run average reward under policy π is given by

$$g_\pi = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}^\pi \left[\sum_{n=1}^{N} r(s_n, a_n) \right]. \tag{15}$$

The limit in the definition of g_π is guaranteed to converge to the same value for any initial state, for a given stationary deterministic policy π, because the model is unichain and has a finite state space.

Let us define a probability distribution that will be useful. Let $\gamma_\pi(k)$ be the long run average fraction of time that there are k customers at Method 1, under policy π. Note γ_π is a probability distribution, with support in $k = 0, 1, ..., M$.

Suppose that for a certain customer class l we have $c_{l1} \geq c_{l2}$, and for all other customer classes $c_{i1} < c_{i2}$. That means that sending a request from class l to Method 2 will result in an immediate cost saving of $c_{l1} - c_{l2}$, compared to sending it to Method 1. Furthermore, if the gatekeeper sends a request from class l to Method 2 the length of the queue will remain unchanged; the next arrival will observe the exact same state as the previous one and the gatekeeper will have the same set of alternatives. However, if the new request is sent to Method 1, then the length of the queue increases by 1. It follows, that for any fixed k, a policy π that always sends class l requests to Method 1 will have a greater $\gamma_\pi(M)$ than one that sends all such requests to Method 2, if the policies are identical in every other state. Hence, the policy π will send arrivals from other customer classes to Method 2 more frequently (when it is in the full state) and this will result in additional costs of $c_{i2} - c_{i1}$ each time it happens.

Because this holds for any k, it follows that $\pi^*(k,l) = 2$ for all k under any optimal policy. This argument extends to the case where there are several customer classes such that $c_{i1} \geq c_{i2}$. Therefore, from now on we assume $c_{i1} < c_{i2}$ for all i without loss of generality.

For any unichain MDP there exists an optimal stationary deterministic policy that satisfies the following optimality equations (see Theorem 8.4.4 of Puterman [58]):

$$w_\pi(s) + g_\pi = \max_{a \in \mathcal{A}_s} \left\{ r(s,a) + \sum_{j \in \mathcal{S}} p(j|s,a) w(s) \right\} \quad \forall s \in \mathcal{S}$$

where w_π is called the bias vector under that policy. Define w_π component-wise as:

$$w_\pi(s) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} (r(s_n, \pi(s_n) - g_\pi) | s_0 = s \right].$$

We are only concerned with the states where more than a single action is available. Let us re-write the optimality equations for states (k,i) such that $k \in \{0,1,2,...,M-1\}, i \in \{1,2\})$ as:

$$w_\pi(k,i) + g = \max \{ -c_{i1} + U_\pi(k+1), -c_{i2} + U_\pi(k) \} \tag{16}$$

where

$$U_\pi(k) = \mu_{1,k} w_\pi((k-1),0) + \sum_{i=1}^{I} \lambda_i w_\pi(k,i) + (m-k)^+ \mu_1 w_\pi(k,0),$$

with $(m-k)^+ = \max(m-k, 0)$. We observe that this model is analogous to an admission control problem presented in Lewis *et al.* [44]. Their results can be applied here with proper modifications. Define $\eta_\pi(s)$ as the long-run average fraction of time the Markov chain spends at state s under policy π. We get the following results.

Proposition 16. *Under any optimal policy π, $U_\pi(k)$ is strictly decreasing, that is*

$$U_\pi(k+1) - U_\pi(k) < 0 \quad \forall k \in (0,1,2,...,M-1).$$

Proof. We use induction on k. For $k = 0$, if $\pi(0,i) = 2$ for some i, then from the optimality equation we can conclude that

$$(c_{i2} - c_{i1}) + U_\pi(k+1) - U(k) \leq 0$$

therefore, since $c_{i1} < c_{i2}$ for all i, we have $U_\pi(k+1) - U_\pi(k) < 0$. So let's assume $\pi(0,i) = 1$ for all i. Then from the definition of $w_\pi(0,0)$

$$U_\pi(0) = \sum_{i=1}^{I} \lambda_i w_\pi(0,i) + \mu_{1,m} w_\pi(0,0) = \sum_{i=1}^{I} \lambda_i w_\pi(0,i) + \mu_{1,m}(U_\pi(0) - g_\pi).$$

Applying the assumption that $\pi(0,i) = 1$ for all i, we get:

$$U_\pi(0) = \sum_{i=1}^{I} \lambda_i(-c_{i1} + U_\pi(1) - g_\pi) + \mu_{1,m}(U_\pi(0) - g_\pi)$$

and applying the fact that $\Lambda + \mu_{1,m} = 1$ we get

$$U_\pi(0) = \sum_{i=1}^{I} \lambda_i(-c_{i1} + U_\pi(1) - g_\pi) + (1 - \Lambda)\,(U_\pi(0) - g_\pi).$$

Hence

$$\sum_{i=1}^{I} \lambda_i(U_\pi(1) - U_\pi(0)) = g_\pi + \sum_{i=1}^{I} \lambda_i c_{i1}.$$

Let R_i be the set of states where customers in class i are sent to Method 1, under the policy π, then:

$$g_\pi = -\sum_{i=1}^{I} \sum_{k:(k,i)\in R_i} \eta(k,i)\lambda_i c_{i1} - \sum_{i=1}^{I} \sum_{k:(k,i)\notin R_i} \eta(k,i)\lambda_i c_{i2}.$$

It follows that:

$$g_\pi + \sum_{i=1}^{I} \lambda_i c_{i1} < 0$$

because $c_{i1} < c_{i2}$ and there must be some states for which we assign arrivals to Method 2. Therefore, $U_\pi(1) - U_\pi(0) < 0$. Now assume the statement holds for $k-1$. Once again if we let $\pi(k,i) = 2$, for some i then the statement follows, from

$$(c_{i2} - c_{i1}) + U_\pi(k+1) - U_\pi(k) \le 0.$$

So, let $\pi(k,i) = 1$ for all i. Here

$$\begin{aligned} U_\pi(k) &= \mu_{1,k} w_\pi((k-1),0) + \sum_{i=1}^{I} \lambda_i w_\pi(k,i) + \mu_{1,(m-k)^+} w_\pi(k,0) \\ &= \mu_{1,k} w_\pi((k-1),0) + \sum_{i=1}^{I} \lambda_i w_\pi(k,i) + (1 - \mu_{1,k} - \Lambda)\,\mu_1 w_\pi(k,0) \end{aligned}$$

applying the the fact that $\pi(k,i) = 1$ for all i, we get

$$\sum_{i=1}^{I} \lambda_i (U_\pi(k+1) - U_\pi(k)) = g_\pi + \sum_{i=1}^{I} \lambda_i c_{i1} + \mu_{1,k}(U_\pi(k) - U(k-1)).$$

Then by induction hypothesis, and the same argument as before we get

$$(U_\pi(k+1) - U_\pi(k)) < 0.$$

■

Define

$$\Delta U_\pi(k) = U_\pi(k+1) - U_\pi(k)$$

and

$$\Delta^2 U_\pi(k) = \Delta(\Delta U_\pi(k)).$$

Similarly define:

$$\Delta w_\pi(k,i) = w_\pi(k+1,i) - w_\pi(k,i),$$

then we can write the following proposition.

Proposition 17. *Under any optimal policy π, $U_\pi(k)$ is strictly concave, that is*

$$\Delta^2 U_\pi(k) < 0 \quad \forall k \in (0, 1, 2, ..., M-2).$$

Proof. We use induction on k. For the base case we show the statement holds for $k = 0$, we have

$$U_\pi(0) = \sum_{i=1}^{I} \lambda_i w_\pi(0,i) + (1 - \Lambda)\, w_\pi(0,0) \tag{17}$$

$$U_\pi(1) = \sum_{i=1}^{I} \lambda_i w_\pi(1,i) + \mu_1 w_\pi(0,0) + (1 - (\Lambda + \mu_1))\, w_\pi(1,0) \tag{18}$$

Recall that $w_\pi(k,0) = U_\pi(k) - g_\pi$, therefore rearranging the above we get

$$\begin{aligned} g_\pi &= \sum_{i=1}^{I} \lambda_i w_\pi(0,i) - \sum_{i=1}^{I} \lambda_i \left(U_\pi(0) - g_\pi\right) \\ &= \sum_{i=1}^{I} \lambda_i w_\pi(1,i) + \mu_1 w_\pi(0,0) - (\Lambda + \mu_1)\left(U_\pi(1) - g_\pi\right). \end{aligned}$$

Hence, we have:

$$\sum_{i=1}^{I} \lambda_i w_\pi(0,i) - \sum_{i=1}^{I} \lambda_i \left(U_\pi(0) - g_\pi\right) = \sum_{i=1}^{I} \lambda_i w_\pi(1,i) + \mu_1 w_\pi(0,0) - (\Lambda + \mu_1)\left(U_\pi(1) - g_\pi\right).$$

Re-arranging a few terms, and using $w_\pi(k,0) = U_\pi(k) - g_\pi$, we get:

$$\sum_{i=1}^{I} \lambda_i (\Delta w_\pi(0,i) - \Delta U_\pi(0)) = \mu_1 \Delta U_\pi(0).$$

Now define the following partition of the customer classes:

1. AA^π = The set of customer classes, such that the given optimal policy π routes them to Method 1, when there are either 0 or 1 customers at Method 1.

2. RR^π = The set of customer classes, such that the given optimal policy π routes them to Method 2, when there are either 0 or 1 customers at Method 1.

3. AR^π = The set of customer classes, such that the given optimal policy π routes them to Method 1, when there are 0 customers at Method 1, and to Method 2, when there is 1 at Method 1.

4. RA^π = The set of customer classes, such that the given optimal policy π routes them to Method 2, when there are 0 customers at Method 1, and to Method 1, when there is 1 at Method 1.

Then we have

$$\begin{aligned}\mu_1 \Delta U_\pi(0) = &\sum_{i \in AA} \lambda_i (\Delta w_\pi(0,i) - \Delta U_\pi(0)) + \sum_{i \in RR} \lambda_i (\Delta w_\pi(0,i) - \Delta U_\pi(0)) \\ &+ \sum_{i \in AR} \lambda_i (\Delta w_\pi(0,i) - \Delta U_\pi(0)) + \sum_{i \in RA} \lambda_i (\Delta w_\pi(0,i) - \Delta U_\pi(0))\end{aligned}$$

Note, from the optimality equation (16) for 0 and 1, using the definition of $U_\pi(0)$ and $U_\pi(1)$ in (17), (18), we can conclude the following:

1. For $i \in AA^\pi$ we have $\Delta w_\pi(0,i) = \Delta U_\pi(1)$.

2. For $i \in RR^\pi$ we have $\Delta w_\pi(0,i) = \Delta U_\pi(0)$.

3. For $i \in AR^\pi$ we have $\Delta w_\pi(0,i) = c_{i1} - c_{i2}$.

4. For $i \in RA^\pi$ we have $\Delta w_\pi(0,i) = c_{i2} - c_{i1} + U_\pi(1)$.

Hence

$$\begin{aligned}
\mu_1 \Delta U_\pi(0) &= \sum_{i \in AA^\pi} \lambda_i(\Delta U_\pi(1) - \Delta U_\pi(0)) + \sum_{i \in AR^\pi} \lambda_i(c_{i1} - c_{i2} - \Delta U_\pi(0)) \\
&\quad + \sum_{i \in RA^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(1)) \\
&= \sum_{i \in AA^\pi} \lambda_i(\Delta^2 U_\pi(0)) + \sum_{i \in AR^\pi} \lambda_i(c_{i1} - c_{i2} - \Delta U_\pi(1)) + \Delta^2 U_\pi(0) \\
&\quad + \sum_{i \in RA^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(1))
\end{aligned}$$

Then it follows that:

$$\begin{aligned}
&\sum_{i \in AA^\pi \cup AR^\pi} \lambda_i(\Delta^2 U_\pi(0)) = \\
&\qquad \mu_1 \Delta U_\pi(0) + \sum_{i \in AR^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(1)) - \sum_{i \in RA^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(1)).
\end{aligned}$$

Consider the right-hand-side of the expression above. This first term is negative because U_π is decreasing. The second term is non-positive, because for the classes in AR^π it is optimal to send arrivals to Method 2 when there is 1 customer at Method 1. Similarly, for the last term, it is optimal to send arrivals from the classes in RA^π to Method 1, that is $c_{i2} - c_{i1} + \Delta U_\pi(1) \geq 0$, when there is 1 customer at Method 1. Then, since U_π is decreasing, we conclude $\Delta^2 U_\pi(0) < 0$.

Now assume the statement holds for $k-1$. Once again we use the definition of U_π, for k and $k+1$ and get

$$\begin{aligned}
g_\pi &= \sum_{i=1}^{I} \lambda_i w_\pi(k,i) + \mu_{1,k} w_\pi(k-1,0) - (\Lambda + \mu_{1,k})\left(U_\pi(k) - g_\pi\right) \\
&= \sum_{i=1}^{I} \lambda_i w_\pi(k+1,i) + \mu_{1,k+1} w_\pi(k,0) - (\Lambda + \mu_{1,k+1})\left(U_\pi(k+1) - g_\pi\right).
\end{aligned}$$

Re-arranging a few terms, and using $w_\pi(k,0) = U_\pi(k) - g_\pi$, we get:

$$\sum_{i=1}^{I} \lambda_i(\Delta w_\pi(k,i) - \Delta U_\pi(k)) = \mathbf{1}_{\{k<m\}}\mu_1 \Delta U_\pi(k) + \mu_{1,k}\Delta^2 U_\pi(k-1)$$

Now define the following partition of the customer classes:

1. AA_k^π = The set of customer classes, such that the given optimal policy π routes them to Method 1, when there are either k or $k+1$ customers at Method 1.

2. RR_k^π = The set of customer classes, such that the given optimal policy π routes them to Method 2, when there are either k or $k+1$ customers at Method 1.

3. AR_k^π = The set of customer classes, such that the given optimal policy π routes them to Method 1, when there are k customers at Method 1, and to Method 2, when there are $k+1$ customers at Method 1.

4. RA_k^π = The set of customer classes, such that the given optimal policy π routes them to Method 2, when there are k customers at Method 1, and to Method 1, when there are $k+1$ customers at Method 1.

Then we have

$$\begin{aligned}
&\mathbf{1}_{\{k<m\}}\mu_1 \Delta U_\pi(k) + \mu_{1,k}\Delta^2 U_\pi(k-1) \\
&\quad = \sum_{i\in AA_k^\pi} \lambda_i(\Delta w_\pi(k,i) - \Delta U_\pi(k)) + \sum_{i\in RR_k^\pi} \lambda_i(\Delta w_\pi(k,i) - \Delta U_\pi(k)) \\
&\qquad + \sum_{i\in AR_k^\pi} \lambda_i(\Delta w_\pi(k,i) - \Delta U_\pi(k)) + \sum_{i\in RA_k^\pi} \lambda_i(\Delta w_\pi(k,i) - \Delta U_\pi(k))
\end{aligned}$$

By a similar argument as before we get

$$\begin{aligned}
&\mathbf{1}_{\{k<m\}}\mu_1 \Delta U_\pi(k) + \mu_{1,k}\Delta^2 U_\pi(k-1) \\
&\quad = \sum_{i\in AA_k^\pi} \lambda_i(\Delta^2 U_\pi(k)) + \sum_{i\in AR_k^\pi} \lambda_i(c_{i1} - c_{i2} - \Delta U_\pi(k+1) + \Delta^2 U_\pi(k)) \\
&\qquad + \sum_{i\in RA_k^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(k+1))
\end{aligned}$$

Then it follows that:

$$\sum_{i \in AA_k^\pi \cup AR_k^\pi} \lambda_i(\Delta^2 U_\pi(k)) = \mathbf{1}_{\{k<m\}}\mu_1 \Delta U_\pi(k) + \mu_{1,k}\Delta^2 U_\pi(k-1)$$
$$+ \sum_{i \in AR_k^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(k+1))$$
$$- \sum_{i \in RA_k^\pi} \lambda_i(c_{i2} - c_{i1} + \Delta U_\pi(k+1)).$$

Consider the right-hand-side of the expression above, by the same reasoning as for the base case, plus using the previous proposition and using the induction hypothesis we can conclude $\Delta^2 U_\pi(k) < 0$. This completes the induction. ■

Theorem 5. *There exists an optimal policy π^* such that: for each customer class i, there exists a threshold k_i^* such that for each $k < k_i^*$ we have $\pi^*(k,i) = 1$ and for each $k \geq k_i^*$ we have $\pi^*(k,i) = 2$.*

Proof. Suppose that π^* is an optimal policy for which the optimality equations hold. Because this is a unichain model with finite state and action spaces such a policy is known to exist. Let $U_{\pi^*}(k)$, be the function $U(k)$ in the optimality equations under π^*. Choose an arbitrary customer class i, and let k_i^* be the smallest k such that $\pi(k,i) = 2$. The optimality equations hold, so

$$(c_{i2} - c_{i1}) + (U_{\pi^*}(k_i^* + 1) - U_{\pi^*}(k_i^*)) \leq 0.$$

By the concavity of U_{π^*} it follows that

$$(c_{i2} - c_{i1}) + (U_{\pi^*}(k+1) - U_{\pi^*}(k)) \leq 0$$

for all $k > k_i^*$. Therefore $\pi^*(k,i) = 2$ for all $k > k_i^*$. Since we chose i arbitrarily, the result holds for each class i. ■

We say a policy π is of pure trunk-reservation form, if

1. There exists a request class i such that $\pi(k,i) = 1$ for $k = 0, 1, ..., M-1$.

2. For each i there exists a threshold k_i such that $\pi(k,i) = 1$ for $k = 0, 1, ..., k_i - 1$ and $\pi(k,i) = 2$ for $k = k_i, ..., M-1$.

3. Furthermore, we say a trunk reservation policy is ordered by b_i, if for some class-specific attribute b_i the following holds: For two customer classes i, l we have $k_i \geq k_l$ if and only if $b_i \leq b_l$.

Corollary 4. *There exists an optimal policy π^* which is of pure trunk-reservation form, ordered by $c_{i1} - c_{i2}$.*

This structure is significant because it is both intuitive and easy to implement. Also, knowing the structure of the policy greatly reduces the universe of potentially optimal policies, making it computationally tractable to solve the problem even for larger instances.

Recall that the system administrator wishes to balance operating cost, security and usability. In this subsection we have not included the latency cost h_{ij}, so this solution only balances the operating cost and the security of the system. It does not take into account usability. If the administrator wants to track the usability measure for a given policy π, the average latency cost experienced by a type i user, can be calculated as:

$$\sum_{k=0}^{M} \gamma_\pi(k) \left(h_{i1} \left(\frac{(k+1-m)^+}{m\mu_1} + \frac{1}{\mu_1} \right) \mathbf{1}_{\{\pi(k,i)=1\}} + \frac{h_{i2}}{\mu_2} \mathbf{1}_{\{\pi(k,i)=2\}} \right).$$

In the next section, we solve the problem under the same capacity assumptions as in this section, but assuming there are hard constraints on the fraction of errors of each type that can be committed.

5.5 Constraint Based Approach

The model presented in the previous section depends heavily on the costs of committing errors, namely, C_I and C_{II}. However, in practice these costs involve several consequences from committing an error and an accurate estimation may be difficult.

On the other hand, subject matter experts may already have in mind a target bound on the fraction of authentications that result in each type of error. In this section, instead of having costs associated with Type-I and Type-II error, we assume that there is a known upper bound on the proportion of requests that suffer Type-I and Type-II errors, called $\bar{\alpha}$ and $\bar{\beta}$, respectively. Now the controller must assign an authentication method to each arriving user request, while ensuring that the long-run fraction of arrivals that result in a Type-I error is below $\bar{\alpha}$ and also that the long-run fraction of arrivals that result in a Type-II error is below $\bar{\beta}$.

As in Section 5.4, we will first analyze an authentication system where each method has infinitely many servers available, and then a system where one method has finite capacity, while all others have infinite capacity. In the second case, we will again simplify the problem by grouping all finite-capacity methods into a single method.

5.5.1 Infinite Capacity for All Methods

Once again, if the capacity available to process each request is very large, then we can assume there are infinitely many servers at each method. So, we treat each method as an $M/G/\infty$ queue. Because all the parameters are stationary, it follows that there exists a stationary policy which is optimal. Furthermore, as in Section 5.4.1, the decision of where to route an incoming request does not depend on the current state of the system and will not affect the cost of future assignments. Therefore, there exists a static policy which is optimal. Hence, we will consider only such policies. Then, we can define a randomized policy π, by defining $q^{\pi}_{i,j}$ as a map from $I \times J$ to $[0, 1]$, where $q^{\pi}_{i,j}$ is the probability that a type i customer is assigned to method j, under policy π. Clearly, we require

$$\sum_{j=1}^{J} q^{\pi}_{ij} = 1 \quad \forall i \in \{1, ..., I\}.$$

Assume (without loss of generality) that $\Lambda = 1$, then the probability that the reward for a given arrival is $-c_j$ is given as: $\sum_{i=1}^{I} \lambda_i q^{\pi}_{ij}$. Then the long run average expected reward per customer of policy π would be:

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[-\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J} c_j \sum_{i=1}^{I}\lambda_i q_{i,j}^{\pi}\right],$$

where $N(t)$ is the counting process of all arrivals. Similarly we can calculate the proportion of arrivals that result in Type-I errors under policy π as

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J}\sum_{i=1}^{I}\lambda_i \alpha_{ij} q_{i,j}^{\pi}\right],$$

and the proportion of Type-II errors is

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J}\sum_{i=1}^{I}\lambda_i \beta_{ij} q_{i,j}^{\pi}\right].$$

The three previous limits all converge, because all the parameters are stationary and bounded, and the counting process $N(t) \to \infty$ with probability 1 as $t \to \infty$.

Therefore, if Π are all the potential policies, we can write the problem as the following optimization problem:

$$(SP1): \qquad \max_{\pi\in\Pi}\left\{-\lim_{t\to\infty} \mathbb{E}^{\pi}\left[\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J} c_j \sum_{i=1}^{I}\lambda_i q_{i,j}^{\pi}\right]\right\}$$

subject to

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J}\sum_{i=1}^{I}\lambda_i \alpha_{ij} q_{i,j}^{\pi}\right] \leq \bar{\alpha}$$

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J}\sum_{i=1}^{I}\lambda_i \beta_{ij} q_{i,j}^{\pi}\right] \leq \bar{\beta}$$

$$\sum_{j=1}^{J} q_{i,j}^{\pi} = 1 \quad \forall\, i = 1, ..., I$$

$$q_{i,j}^{\pi} \geq 0 \quad \forall\, i = 1, ..., I,\ j = 1, ..., J$$

Proposition 18. *There exists an optimal policy π^* for $(SP1)$, which is randomized and static. Assume $\Lambda = 1$, then such a policy can be determined by solving the* *following linear program*

$$(LP1): \quad \max\left\{-\sum_{j=1}^{J}\sum_{i=1}^{I} c_j\lambda_i q_{ij}\right\}$$

subject to

$$\sum_{j=1}^{J}\sum_{i=1}^{I} \lambda_i\alpha_{ij}q_{ij} \leq \bar{\alpha}$$

$$\sum_{j=1}^{J}\sum_{i=1}^{I} \lambda_i\beta_{ij}q_{ij} \leq \bar{\beta}$$

$$\sum_{j=1}^{J} q_{ij} = 1 \quad \forall i = 1, ..., I$$

$$q_{ij} \geq 0 \quad \forall i = 1, ..., I,\ j =, ..., J$$

and defining the policy π^* *as follows: let* q^* *be an optimal solution to* $(LP1)$*, whenever the controller receives a type* i *request, he should assign it to method* j *with probability* q^*_{ij}.

Proof. The problem of determining an optimal policy consists of determining probabilities $q_{ij}, \forall i \in \{1, ..., I\}, j \in \{1, ..., J\}$, where q_{ij} is a decision variable, representing the probability of assigning each incoming request from class i to authentication method j.

Since the objective (the reward) and the constraints (fraction of errors) of the optimization problem are expressed in terms of expectations, and the state and action spaces are finite, those expectations can be expressed as finite sums, we can formulate it as a deterministic Linear Program (LP). Consider the arrivals of all requests, which by assumption follow a Poisson process with rate Λ. Each arrival collects a reward which is i.i.d. We consider costs as negative rewards, as in previous sections. Therefore, we have a Poisson process, where each arrival has bounded i.i.d. rewards. This is a renewal-reward process. For this type of process it is known (see, for example, Proposition 3.4.1 in Resnick [60]) that the long run average reward is equal to the expectation of a single reward, that is:

$$\lim_{t\to\infty} \mathbb{E}^{\pi}\left[-\frac{1}{N(t)}\sum_{n=0}^{N(t)}\sum_{j=1}^{J} c_j \sum_{i=1}^{I} \lambda_i q_{i,j}^{\pi}\right] = -\sum_{j=1}^{J} c_j \sum_{i=1}^{I} \lambda_i q_{ij}.$$

By the same argument the long run average fraction Type-I errors is equal to the probability that a single arrival will result in a Type-I error; and similarly for Type-II errors. Therefore, we can express the stochastic optimization problem $(SP1)$ as $(LP1)$. So a solution to $(LP1)$ determines an optimal policy for $(SP1)$. ■

In this case we can further capture the three-way trade-off between usability, cost and security by adding the following constraint to the $(LP1)$ model:

$$\sum_{j=1}^{J}\sum_{i=1}^{I} \lambda_i q_{ij} \frac{h_{ij}}{\mu_j} \leq \bar{h} \tag{19}$$

where $\bar{h}$ is an upper bound on the average latency cost experienced by all users. Call the new LP which includes constraint (19), $(LP1')$. Appropriately adjusting $\bar{\alpha}$, $\bar{\beta}$ and $\bar{h}$ allows the system administrator to give more priority to security, usability or operating cost. Section 5.6 uses a numerical example to illustrate how a system administrator can use $(LP1')$ to build an efficient frontier of non-dominated solutions to choose from.

5.5.2 One Method with Finite Capacity

In this section, we analyze the case where one method has finite capacity and one method has infinite capacity. Following the same procedure as in the cost-based approach, the results presented here can be extended to the case where there are multiple methods with infinite capacity. Assume Method 1 has m parallel identical servers, with exponentially distributed service times, and a total finite capacity M and Method 2 has infinitively many parallel identical servers. This is the same situation as depicted in Figure 5.4.2.

This problem can be reduced to an admission control problem to Method 1 with constraints. Fan-Orzechowski and Feinberg [18] solve an admission control problem, which is analogous to this one. In their setup a reward is collected for each admission and costs of several types are incurred for each rejection. There is a constraint on how much of each cost is incurred on average. The problem presented here is similar to that one, but differs in a few aspects. First, rewards (negative costs) are collected for both admissions and rejection. Similarly, penalties are incurred not only when there is a rejection (corresponding to using Method 2), but also when there is admission (corresponding to using Method 1). Finally, our model is also different in the sense that the penalties are not monetary costs, but a measure of the risk of committing an error. However, if we define $r := c_2 - c_1$, then r can be considered a reward gained each time we assign a request to Method 1. Similarly, we can define $\alpha_{i2} - \alpha_{i1}$ and $\beta_{i2} - \beta_{i1}$ as penalties for the Type-I and Type-II fraction of errors, respectively. In this case our model is equivalent to the one presented in Fan-Orzechowski and Feinberg [18]. We note that those authors only consider costs in the constraints, but their results hold for any constraint which is a weighted average of the state-action frequencies, as evidenced by our application.

Fan-Orzechowski and Feinberg [18] model a similar parllel queueing system as a semi-Markov process. We take an alternative approach and model the problem as an MDP with constraints, using uniformization. First, we model the unconstrained problem as an MDP, then we add the constraints. Note that, for the unconstrained problem we can use the same MDP formulation described in Section 5.4.2, except that the rewards for this model are different.

Hence we get the same uniformization constant $\Lambda + \mu_{1,m} < \infty$, the same state space $\mathcal{S}$ and the same actions for each state $\mathcal{A}_s$. We assume (without loss of generality) that $\Lambda = 1$. Then, we get the same transition probabilities $p(s'|s,a)$, as defined in Section 5.4.2, except each probability needs to be divided by $\Lambda + \mu_{1,m}$. In this case, because we are modeling the problem as admission control into Method 1 we define the rewards differently. When determining the reward of assigning a request to

Method 1 or Method 2 (actions 1 and 2), we define the reward as the relative *savings* compared to assigning every incoming request to Method 2. When the last event was a service completion, the only available action is 0 or *do nothing*, clearly the 0 action has 0 reward. Hence, the rewards are given by:

$$
\begin{aligned}
&r((k,i),0) = 0 && \forall i \in \{1,...,I\}, 0 \le k \le M,\\
&r((k,i),1) = c_2 - c_1 = r && \forall i \in \{1,...,I\}, 0 \le k \le M-1,\\
&r((k,i),2) = c_2 - c_2 = 0 && \forall i \in \{1,...,I\}, 0 \le k \le M.
\end{aligned}
$$

This is a unichain MDP, by the same argument as in Section 5.4.2. Because all the parameters are stationary, the MDP is unichain and the state and action spaces are finite, there exists a stationary deterministic policy which is optimal. This implies that there exist stationary randomized policies which are optimal, since deterministic policies are a subset of randomized policies.

We define a stationary randomized policy π, using a mapping $y_\pi : \{s = (k,i) \in \mathcal{S} : i \neq 0\, k \neq M\} \to [0,1]$, where the policy π is defined as follows:

1. If a request of type i arrives when there are k customers at Method 1, assign it to Method 1 with probability $y_\pi(k,i)$.

2. If a request of type i arrives when there are k customers at Method 1, assign it to Method 2 with probability $1 - y_\pi(k,i)$.

3. If a request of type i arrives when there are M customers at Method 1, assign it to Method 2.

4. If there is a real or fictitious service completion, do nothing.

Note that for any possible policy π, there exists a y_π that fully determines it. Our objective is to find a policy π^* which maximizes the long run average gain g_π, defined in (15). The limit in (15) is guaranteed to converge to the same value for any initial state, for a given stationary randomized policy π, because the model is unichain and has a finite state space.

In this model, we observe that under a given policy π the number of customers at Method 1 can be modeled as a Markov chain $\{Z_n, n = 0, 1, ...\}$. Define $\gamma_\pi(k)$ (as in section 5.4.2) as the long run fraction of time there are k customers in the system under policy π. Now, following a similar argument as in Proposition 18 we can re-write the gain g_π under a stationary randomized policy π, as

$$g_\pi = r \sum_{i=1}^{I} \lambda_i \sum_{k=0}^{M-1} y_\pi(k,i)\gamma_\pi(k), \tag{20}$$

where y_π and γ_π satisfy the following balance and normalization equations:

$$\sum_{i=1}^{I} \lambda_i y_\pi(k,i)\gamma_\pi(k) = \mu_{1,k+1}\gamma_\pi(k+1) \quad \forall k = 0, 1, ..., M-1 \tag{21}$$

$$\sum_{k=0}^{M} \gamma_\pi(k) = 1 \tag{22}$$

$$\gamma_\pi(k) \geq 0 \quad \forall k = 0, 1, ..., M. \tag{23}$$

Here, because the resulting Markov chain is unichain, aperiodic and has finite state space, it follows that for a given policy π, with a specific y_π there exists a unique solution γ_π to (21), (22) and (23).

By the above argument, finding a policy π that maximizes g_π is equivalent to finding a pair (y_π, γ_π), which maximize (20), subject to (21), (22) and (23). Since y_π and γ_π are both variables this is not an LP. But we can apply a change of variables, defining $x_\pi(k,i) = y_\pi(k,i)\gamma_\pi(k)$ and formulate the following linear program (we drop the π sub-indices for convenience):

$$(LP2): \quad \max \left\{ r \sum_{i=1}^{I} \lambda_i \sum_{k=0}^{M-1} x(k,i) \right\}$$

subject to

$$\sum_{i=1}^{I} \lambda_i x(k,i) = \mu_{1,k+1}\gamma(k+1) \quad \forall k = 0,1,...,M-1$$

$$\sum_{k=0}^{M} \gamma(k) = 1$$

$$0 \leq x(k,i) \leq \gamma(k) \quad \forall i = 1,...,I,\ k = 0,1,...,M-1.$$

In $(LP2)$ the decision variables are x and γ.

Proposition 19. *Let* (x^*, γ^*) *be an optimal solution to* $(LP2)$ *then setting*

$$y_{\pi^*}(k,i) = \begin{cases} \dfrac{x^*(k,i)}{\gamma^*(k)} & \gamma^*(k) > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{24}$$

defines a policy π^* *which is optimal for the unconstrained MDP.*

Proof. We proceed by contradiction. Suppose there exists another policy π' that is feasible and performs better than π^* for the unconstrained MDP. Then there exists a pair $(y_{\pi'}, \gamma_{\pi'})$ such that (21), (22) and (23) hold. Set $x'(k,i) = y_{\pi'}(k,i)\gamma_{\pi'}(k,i)$ for each (k,i). Then the pair $(x', \gamma_{\pi'})$ is feasible for $(LP2)$ and performs better than (x^*, γ^*). This contradicts the optimality of (x^*, γ^*). So π^* is optimal for the unconstrained MDP. ■

Now we consider the constrained MDP. By a similar argument as in Proposition 18 we can formulate the constraint on the long-run fraction of requests that result in Type-I and Type II errors under policy π, respectively as

$$\sum_{i=1}^{I} \lambda_i \left[\alpha_{i2}(1 - \sum_{k=0}^{M-1} y_\pi(k,i)\gamma(k)) + \alpha_{i1} \sum_{k=0}^{M-1} y_\pi(k,i)\gamma(k) \right] \leq \bar{\alpha} \tag{25}$$

$$\sum_{i=1}^{I} \lambda_i \left[\beta_{i2}(1 - \sum_{k=0}^{M-1} y_\pi(k,i)\gamma(k)) + \beta_{i1} \sum_{k=0}^{M-1} y_\pi(k,i)\gamma(k) \right] \leq \bar{\beta}. \tag{26}$$

Hence we define the constrained MDP as maximizing (20), subject to (21), (22), (23), (25) and (26). Once again we use the $x(k,i)$ variables to formulate a linear program. Suppose (x,γ) is a solution to $(LP2)$ then the sum $\sum_{k=0}^{M-1} x(k,i)$ represents the long-run fraction of type i arrivals that are assigned to Method 1. Conversely $1-\sum_{k=0}^{M-1} x(k,i)$ is the long-run fraction of type i arrivals that are assigned to Method 2. Then, we can write a new LP, which considers the constraints on the long-run fraction of errors as:

$$(LP3): \quad \max\left\{ r\sum_{i=1}^{I}\lambda_i \sum_{k=0}^{M-1} x(k,i)\right\}$$

subject to

$$\sum_{i=1}^{I}\lambda_i\left[\alpha_{i2}\left(1-\sum_{k=0}^{M-1}x(k,i)\right)+\alpha_{i1}\sum_{k=0}^{M-1}x(k,i)\right]\le\bar{\alpha} \tag{27}$$

$$\sum_{i=1}^{I}\lambda_i\left[\beta_{i2}\left(1-\sum_{k=0}^{M-1}x(k,i)\right)+\beta_{i1}\sum_{k=0}^{M-1}x(k,i)\right]\le\bar{\beta} \tag{28}$$

$$\sum_{i=1}^{I}\lambda_i x(k,i)=\mu_{1,k+1}\gamma(k+1) \quad \forall k=0,1,...,M-1$$

$$\sum_{k=0}^{M}\gamma(k)=1$$

$$0\le x(k,i)\le\gamma(k) \quad \forall i=1,...,I,\ k=0,1,...,M-1. \tag{29}$$

In $(LP3)$ the decision variables are again x and γ.

Proposition 20. *Let (x^*,γ^*) be an optimal solution to $(LP3)$ then setting y_{π^*} as in (24) defines a policy π^* which is optimal for the constrained MDP.*

Proof. We proceed by contradiction. Suppose there exists another policy π' that is feasible for the constrained MDP and performs better than π^*. Then there exists a pair $(y_{\pi'},\gamma_{\pi'})$ such that (21), (22), (23), (25) and (26) hold. Set $x'(k,i) = y_{\pi'}(k,i)\gamma_{\pi'}(k,i)$ for each (k,i). Then the pair $(x',\gamma_{\pi'})$ is feasible for $(LP3)$ and performs better than (x^*,γ^*). This contradicts the optimality of (x^*,γ^*). So π^* is optimal for the constrained MDP. ■

Solving ($LP3$) using a standard LP solver, provides an easy and efficient way of determining if the constrained MDP is feasible and obtaining an optimal policy. It is not obvious from the parameters whether ($LP3$) is feasible or not. The following condition:

$$\min\{\alpha_{i1}, \alpha_{i2},\} \leq \bar{\alpha},\ \min\{\beta_{i1}, \beta_{i2}\} \leq \bar{\beta}\ \forall i = 1, ..., I$$

is clearly necessary, but not sufficient for ($LP3$) to be feasible. On the other hand, the following condition:

$$\max\{\alpha_{i1}, \alpha_{i2},\} \leq \bar{\alpha},\ \max\{\beta_{i1}, \beta_{i2}\} \leq \bar{\beta}\ \forall i = 1, ..., I$$

implies that every policy is feasible and the problem has a trivial solution. That is, to assign requests to to the cheapest method whenever capacity is available and to the other method, otherwise.

We are further interested on whether the optimal policies obtained using this method have a particular structure. Specifically, we are interested in whether under certain conditions the optimal policy is monotone in some sense, and on exactly how many states the decision is randomized. Namely, (see Theorem 1 of Ross [61]) for stationary finite-state and finite-action unichain MDPs, with n inequality constraints over the state-action frequencies, there exists a so-called n-randomized stationary policy which is optimal. This is a stationary policy that is deterministic in all but n states. We would like to determine a way to find such a policy.

Recall that for the cost based approach the structure of the optimal policy depends on how the two methods compare to each other. Under that approach we could compare the overall performance of the methods for class i using the single parameter c_{ij}. In particular, for each class i such that $c_{i2} \leq c_{i1}$, it is optimal (under the cost-based approach) to assign requests to Method 2, and for the remaining classes there exists an optimal policy which is of trunk reservation form. In this case, however, the methods cannot be compared by a single aggregated parameter. For example, one method may be better at Type-I errors, while the other is better at Type-II errors. Thus, the structure of the optimal policy depends on how the methods compare. There are a total of eight cases, and all of them are feasible in practice. We focus on the cases where one of the methods dominates the other in every way. That is, c_j, α_j and β_j are all lower for one of the methods. These cases are the most mathematically tractable and they are not uncommon in practice. An example of this is the process to unlock a smart-phone. The user can input a PIN, which has low Type-I and Type-II error and uses few resources; or he can use a novelty method, such as facial-recognition, which is resource intensive and is still under development, and as such is error-prone. The remaining cases are left as future work. However, we note that the results presented here may hold under less restrictive assumptions.

5.5.2.1 *Case 1: Method 2 is Cheaper and More Reliable*

In this case, the problem is feasible if, and only if:

$$\alpha_{i2} \leq \bar{\alpha},\ \beta_{i2} \leq \bar{\beta}\ \forall i = 1, ..., I \tag{30}$$

as otherwise one of the security constraints would never hold. If the problem is feasible it is intuitive that the optimal solution is deterministic, and it is to send all authentication requests to Method 2. Here we prove this formally.

Proposition 21. *Assume $c_1 > c_2$, $\alpha_1 > \alpha_2$ and $\beta_1 > \beta_2$. Assume also that the constrained MDP is feasible. Then a policy π with $y_\pi(k, i) = 0\ \forall k, i$ (that is a policy that sends all requests to Method 2 with probability 1) is optimal.*

Proof. If $c_1 > c_2$ then it is obvious that the objective (20) is bounded above by 0. Setting $y_\pi(k, i) = 0\ \forall k, i$, implies (by (21)) that $\gamma_\pi(k) = 0$ for all $k = 1, 2, ..., M$. This in turn implies (by (22)) that $\gamma_\pi(0) = 1$. So (21), (22) and (23) all hold. Also, under this policy constraints (25) and (26) become:

$$\sum_{i=1}^{I} \lambda_i \alpha_{i2} \leq \bar{\alpha}$$
$$\sum_{i=1}^{I} \lambda_i \beta_{i2} \leq \bar{\beta}.$$

By condition (30), and the assumption that $\Lambda = 1$, these constraints hold if the problem is feasible. Also, $y_\pi(k,i) = 0$, means that the objective value (20) of this policy is 0. Since this policy is feasible and achieves the upper bound, then it is optimal. ■

5.5.2.2 Case 2: Method 1 is Cheaper and More Reliable

Here we assume $c_1 < c_2$, $\alpha_1 < \alpha_2$ and $\beta_1 < \beta_2$. Therefore, it is desirable to send all requests to Method 1, but this is not always possible, due to its finite capacity. In this case the following condition:

$$\alpha_{i1} \leq \bar{\alpha},\ \beta_{i1} \leq \bar{\beta}\ \forall i = 1, ..., I$$

is necessary for feasibility. However, since Method 1 has finite capacity, this condition is not sufficient. In general, it is not possible to determine the feasibility of this problem *a priori* based on the parameters. We assume that $(LP3)$ is feasible for the given parameters.

Note that the problem $(LP3)$ is bounded, so if it is feasible it has an optimum. Consider an arbitrary optimal solution to the dual of $(LP3)$ and let ν_1, ν_2 be the dual variables associated with the two error constraints (27) and (28). We can write a Lagrangian relaxation of $(LP3)$ by dualising those two constraints with their corresponding Lagrange multipliers. The objective function of the relaxation would be:

$$\begin{aligned} r \sum_{i=1}^{I} \lambda_i \sum_{k=0}^{M-1} x(k,i) - \nu_1 \left(\sum_{i=1}^{I} \lambda_i \left[\alpha_{i2} \left(1 - \sum_{k=0}^{M-1} x(k,i) \right) + \alpha_{i1} \sum_{k=0}^{M-1} x(k,i) \right] - \bar{\alpha} \right) \\ - \nu_2 \left(\sum_{i=1}^{I} \lambda_i \left[\beta_{i2} \left(1 - \sum_{k=0}^{M-1} x(k,i) \right) + \beta_{i1} \sum_{k=0}^{M-1} x(k,i) \right] - \bar{\beta} \right). \end{aligned}$$

Define an adjusted reward for each customer class as:

$$r'_i := (c_2 - c_1) + \nu_1(\alpha_{i2} - \alpha_{i1}) + \nu_2(\beta_{i2} - \beta_{i1}).$$

Note that, under our assumptions, each term in this expression is non-negative, so the adjusted rewards are non-negative. Furthermore, each r'_i is different, because by assumption the impostor probabilities of each class are different. Reordering some terms in the objective function we get the following Lagrangean relaxation of $(LP3)$:

$$(LR1): \quad \max \left\{ \sum_{i=1}^{I} \lambda_i r'_i \sum_{k=0}^{M-1} x(k,i) - \nu_1 \left(\sum_{i=1}^{I} \lambda_i \alpha_{i2} - \bar{\alpha} \right) - \nu_2 \left(\sum_{i=1}^{I} \lambda_i \beta_{i2} - \bar{\beta} \right) \right\}$$

subject to

$$\sum_{i=1}^{I} \lambda_i x(k,i) = \mu_{1,k+1} \gamma(k+1) \quad \forall k = 0,1,...,M-1$$

$$\sum_{k=0}^{M} \gamma(k) = 1$$

$$0 \leq x(k,i) \leq \gamma(k) \quad \forall i = 1,...,I,\ k = 0,1,...,M-1.$$

Note that $(LR1)$ is itself also a linear program.

Lemma 4. *Any basic solution for $(LP3)$ is optimal for $(LR1)$ and has the same objective function value.*

Proof. Let $((x^*, \gamma^*), \nu^*)$ be an optimal primal-dual pair of solutions to $(LP3)$ and its dual. Because the solution (x^*, γ^*) is feasible for $(LP3)$ it is also feasible for $(LR1)$. We know that ν_1 and ν_2 in the objective function of $(LR1)$ come from a optimal solution to the dual of $(LP3)$ (not necessarily ν^*), then it follows by complementary slackness (see for example Theorem 4.5 in Bertsimas and Tsitsiklis [6]) that

$$\nu_1 \left(\sum_{i=1}^{I} \lambda_i \left[\alpha_{i2} \left(1 - \sum_{k=0}^{M-1} x^*(k,i) \right) + \alpha_{i1} \sum_{k=0}^{M-1} x^*(k,i) \right] - \bar{\alpha} \right) = 0$$

$$\nu_2 \left(\sum_{i=1}^{I} \lambda_i \left[\beta_{i2} \left(1 - \sum_{k=0}^{M-1} x^*(k,i) \right) + \beta_{i1} \sum_{k=0}^{M-1} x^*(k,i) \right] - \bar{\beta} \right) = 0$$

as the expression in parenthesis represent the slacks of the constraints corresponding to dual variables ν_1 and ν_2. This shows that if (x^*, γ^*) is optimal for $(LP3)$ then it has the same objective value in $(LR1)$. Because the pair $((x^*, \gamma^*), \nu^*)$ is primal and dual optimal, it meets the Karush-Kuhn-Tucker (KKT) conditions (see Theorem 12.1 in Nocedal [54]) for $(LP3)$ and its dual. The KKT conditions for $(LR1)$ are a subset of the KKT conditions for $(LP3)$, as it has the same objective function value for (x^*, γ^*) and a subset of the constraints. So the pair $((x^*, \gamma^*), \nu^*)$ also meets the KKT conditions for $(LR1)$, hence (x^*, γ^*) is also optimal for $(LR1)$. ■

We will use this result to prove the main result in this section. Define $\lfloor x \rfloor$ as the integer part of x. We say a policy π is of 2-randomized trunk-reservation form, ordered by r'_i if

1. There exists a request class i such that $y_\pi(k,i) = 1$ for $k = 0, 1, ..., M-1$.
2. For each i there exists a threshold k_i, which may not be an integer such that $y_\pi(k,i) = 1$ for $k = 0, 1, ..., \lfloor k_i \rfloor - 1$, $y_\pi(k,i) = 0$ for $k = \lfloor k_i \rfloor + 1, ..., M-1$, and $y_\pi(\lfloor k_i \rfloor, i) = k_i - \lfloor k_i \rfloor$.
3. All, except at most 2 thresholds k_i are integers.
4. For two customer classes i, l we have that $r'_i > r'_l$ implies $k_i \geq k_l$.

Theorem 6. *If the constrained MDP is feasible, there exists a 2-randomized optimal policy which is a 2-randomized trunk reservation policy, ordered by r'_i. Furthermore, taking any basic optimal solution to $(LP3)$, (x^*, γ^*) and setting*

$$y_{\pi^*}(k,i) = \frac{x^*(k,i)}{\gamma^*(k)}$$

defines an optimal 2-randomized trunk reservation policy π^* for the constrained MDP.

Before we proceed with the proof, we highlight that the importance of this result is three-fold. First, it describes the structure of the optimal policy, which dramatically reduces the search space, when looking for the optimum. Second, this structure is intuitive and easy to implement in practice as a simple set of rules. And lastly, the final statement of the theorem provides a practical mean for finding an optimal policy with the desired structure.

We should also mention, as in previous sections, we are not including the latency cost as part of the optimization process. However, for a given policy the average latency cost experienced by a type i user, under π can be calculated as

$$\sum_{k=0}^{M} \gamma_\pi(k) \left(h_{i1} \left(\frac{(k+1-m)^+}{\mu_{1,m}} + \frac{1}{\mu_1} \right) y_\pi(k,i) + \frac{h_{i2}}{\mu_2} (1 - y_\pi(k,i)) \right).$$

The proof for Theorem 6 follows from Fan-Orzechowski and Feinberg [18]. In order to proceed with the proof we introduce some intermediate results.

Lemma 5. *If* (x^*, γ^*) *is optimal for* $(LR1)$ *then* $\gamma^*(k) > 0$, $\forall k = 0, 1, ..., M-1$.

For a proof we refer the reader to the proof of Lemma 3.4 in Fan-Orzechowski and Feinberg [17].

Proposition 22. *Taking any basic optimal solution to* $(LP3)$, (x^*, γ^*) *and setting*

$$y_{\pi^*}(k,i) = \frac{x^*(k,i)}{\gamma^*(k)}$$

results in a 2-randomized stationary optimal policy for the constrained MDP.

Proof. In Proposition 20, we showed that if (x^*, γ^*) is an optimal solution to $(LP3)$ then setting y_π as above defines a policy π which is optimal for the constrained MDP. Consider the $(LP3)$ in standard form adding non-negative slack variables S_1, S_2 to (27) and (28) respectively, as well as $z(k,i)$ to each constraint (29). We get the following LP:

$$(LP3') : \quad \max \left\{ (c_2 - c_1) \sum_{i=1}^{I} \lambda_i \sum_{k=0}^{M-1} x(k,i) \right\}$$

subject to

$$\sum_{i=1}^{I} \lambda_i \left[\alpha_{i2} \left(1 - \sum_{k=0}^{M-1} x(k,i) \right) + \alpha_{i1} \sum_{k=0}^{M-1} x(k,i) \right] + S_1 = \bar{\alpha}$$

$$\sum_{i=1}^{I} \lambda_i \left[\beta_{i2} \left(1 - \sum_{k=0}^{M-1} x(k,i) \right) + \beta_{i1} \sum_{k=0}^{M-1} x(k,i) \right] + S_2 = \bar{\beta}$$

$$\sum_{i=1}^{I} \lambda_i x(k,i) = \mu_{1,k+1} \gamma(k+1) \quad \forall k = 0, 1, ..., M-1$$

$$\sum_{k=0}^{M} \gamma(k) = 1$$

$$x(k,i) + z(k,i) = \gamma(k) \quad \forall i = 1, ..., I, \; k = 0, 1, ..., M-1$$

$$x(k,i) \geq 0, \quad z(k,i) \geq 0 \quad \forall i = 1, ..., I, \; k = 0, 1, ..., M-1.$$

The resulting standard form LP has $(M \times (I+1)+3)$ constraints and $(2M \times I + I + 3)$ variables. So each basic feasible solution will have at most $(M \times (I+1)+3)$ non-zero variables. By Lemma 5 it follows that $\gamma(k) > 0$ for the $(M+1)$ variables $\gamma(k), \; k = 0, ..., M$. So at most $(M \times I + 2)$ basic variables between $x(k,i)$ and $z(k,i)$ can be non-zero. Because $x(k,i) + z(k,i) = \gamma(k) > 0$, then for each pair (k,i) at least one of the variables must be positive. Therefore, in at most two cases $x(k,i)$ and $z(k,i)$ can both take positive values. By the definition of y_{π^*}, it follows that in all, but at most 2 cases $y_{\pi^*}(k,i)$ is equal to either 1 or 0. So π^* is a 2-randomized optimal policy for the constrained MDP. ■

The following result follows directly from Theorem 3.1 in Feinberg and Reiman [19]. We refer the reader to that reference for a proof. A variation on this result is presented in both Fan-Orzechowski and Feinberg [17] and Fan-Orzechowski and Feinberg [18].

Proposition 23. *Consider any optimal solution* (x^*, γ^*) *to* $(LR1)$*, and define a randomized stationary optimal policy* π *as* $y_\pi(k,i) = \frac{x^*(k,i)}{\gamma^*(k)}$*. Then for an unconstrained MDP with rewards* $r((k,i),1) = r'_i\ \forall k = 0,1,...,M-1,\ i = 1,...,I$ *and 0 otherwise, we have the following:*

1. *For any* i, l*, such that* $r'_i > r'_l$ *,*

$$y_\pi(k,i) \geq y_\pi(k,l) \quad \forall k = 0,...,M-1, \quad i,l = 1,2,...,I.$$

2. *For each* $k = 0,...,M-1$*, all the probabilities* $y_\pi(k,i),\ i = 1,...,I$ *except at most one, are equal to either 0 or 1.*

3. *For a request type* l *such that* $r'_l = \max_i\{r'_i\}$ *we have* $y_\pi(k,l) = 1,\ \forall k = 0,1,...,M-1.$

4. $$y_\pi(k,i) \geq y_\pi(k+1,i) \quad \forall k = 0,...,M-1, \quad i = 1,2,...,I.$$

   and for each $i = 1,...,I$ *all the probabilities* $y_\pi(k,i),\ k = 0,1,...,M-1$ *except at most one, are equal to either 0 or 1.*

This implies that π *is a randomized trunk reservation policy, where the decision will be randomized in at most* $(I-1)$ *states.*

With this we can complete the proof for Theorem 6.

Proof of Theorem 6. We can see that π^* is properly defined, because of Lemma 5. Then, from Propositions 20 and 22 we have that any optimal basic solution of $(LP3)$ produces a 2-randomized policy for the constrained MDP. By Lemma 4, (x^*, γ^*) is also optimal for $(LR1)$. By Proposition 23 any optimal solution to $(LR1)$ defines an $(I-1)$-randomized trunk reservation policy ordered by its rewards r'_i for the unconstrained MDP with the adjusted rewards. Then, putting all together we get that π^* is an optimal 2-randomized trunk reservation policy for the constrained MDP, which is ordered by the adjusted rewards r_i' . ■

Note that during the proof of Theorem 6 we never used the condition that $c_1 < c_2$, $\alpha_1 < \alpha_2$ and $\beta_1 < \beta_2$, except to guarantee the $r_i' > 0$ for each class. Therefore, we conclude that this a sufficient, but not a necessary condition for Theorem 6 to hold. In fact, the necessary condition is simply $r_i' > 0$ for each i. However, this condition can not be verified *a priori* as it includes dual optimal variables of $(LP3)$. But, we conclude that the structure of the optimal policy described in Theorem 6 holds for a broader class of instances, not limited to $c_1 < c_2$, $\alpha_1 < \alpha_2$ and $\beta_1 < \beta_2$.

5.6 Numerical Illustration

In this section we provide a numerical example, which illustrates how a system designer can use the models developed here to balance the three-way trade-off between usability, operating cost and security. Consider an authentication system with four authentication methods and four request classes. Assume that each authentication method has infinitely many servers. Suppose that the application requires setting strict limits on the fraction of errors and the latency cost. In this case we can apply the model $(LP1')$ described in Section 5.5.1. The controller needs to solve $(LP1')$ to find the optimal assignment policy for a fixed set of upper bound parameters $\bar{\alpha}, \bar{\beta}, \bar{h}$.

In order to simplify the analysis, assume that the system is not concerned about Type-I errors, equivalently, we can set $\bar{\alpha} = 1$. So, the only measure of security in this case is the fraction of Type-II errors. We show how changing the bounds on error $(\bar{\beta})$ and on latency cost $(\bar{h})$, either individually or simultaneously affect all three measures of performance. Let $p = [0.2, 0.4, 0.6, 1.0]$, $c = [4, 40, 20, 80]$, $\lambda = [0.25, 0.25, 0.25, 0.25]$, $\mu = [1, 5, 10, 20]$, $\beta = [0.02, 0.2, 0.5, 0.2]$. And set $h_{ij} = h_i$, where $h = [40, 20, 8, 4]$.

First, we study how the constraint on latency cost affects operating cost. Fix $\bar{\beta}$

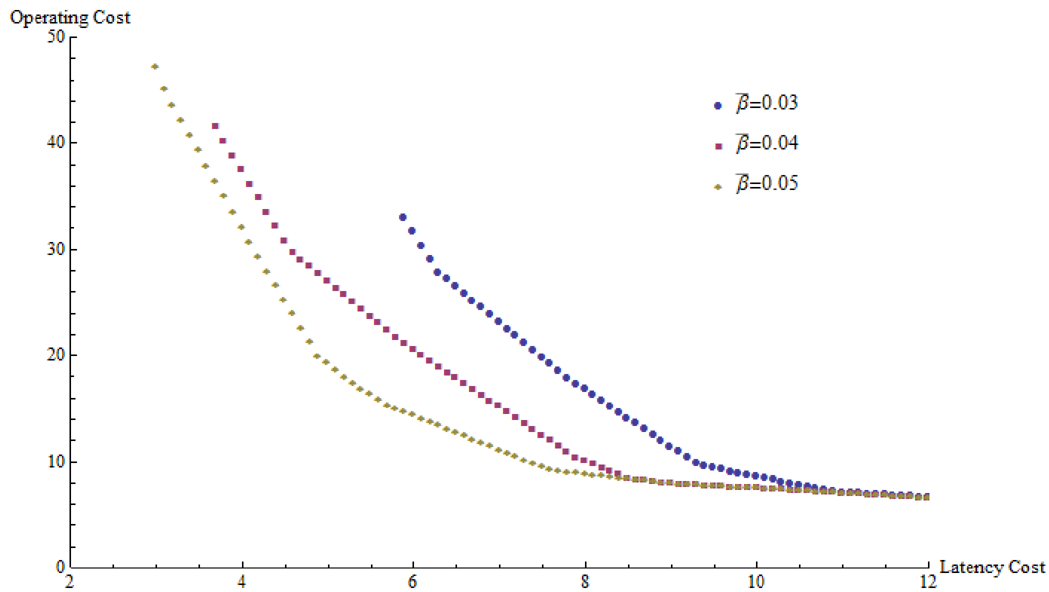

Figure 12: Operating cost of non-dominated solutions for fixed values of $\bar{\beta}$

and let $\bar{h}$ vary between 0 and 12. Figure 12 presents the optimal cost for each value of $\bar{h}$ under fixed $\bar{\beta}$, each data series corresponds to different value of $\bar{\beta}$. Note that as we allow usability to deteriorate ($\bar{h}$ increases), the optimal operating cost improves. Also, that as we require a more secure system (reduce $\bar{\beta}$) the minimum operating cost for a given usability level increases. For smaller values of $\bar{h}$ than the ones presented in Figure 12 $(LP1')$ is infeasible.

Now, consider the effect of the Type-II error constraint on cost while fixing $\bar{h}$. We calculate the value of the optimal policy varying $\bar{\beta}$ between 0 and 0.12, for three fixed values of $\bar{h}$. The results are presented in Figure 13. We observe that as we allow security to deteriorate ($\bar{\beta}$ increases) the optimal operating cost improves. Also, as the bound on average latency cost becomes more strict (reduce $\bar{h}$) the minimum operating cost for a given security level increases. For smaller values of $\bar{\beta}$ than the ones presented in Figure 13 $(LP1')$ is infeasible.

Finally, we calculate the value of the optimal policy while varying the values of $\bar{\beta}$ and $\bar{h}$ simultaneously. Figure 14 shows the efficient frontier of solutions. Each point in Figure 14 represents a non-dominated optimal solution to $(LP1')$, for a

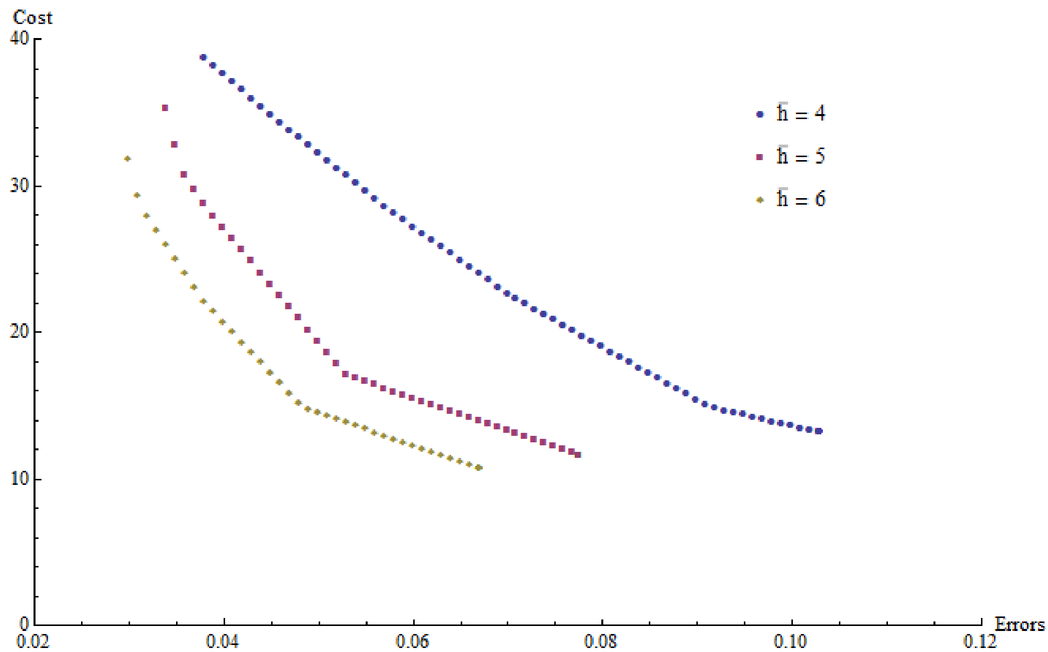

Figure 13: Operating cost of non-dominated solutions for fixed values of $\bar{h}$

given value of $\bar{\beta}$ and $\bar{h}$. That is, for each point in Figure 14, the system designer cannot improve any of the objectives, without deteriorating another one. Each point's coordinates in the graph are: (average latency cost, long-run fraction of Type-II errors, minimal operating cost). A system designer can use this efficient frontier to select the parameters that best balance the three-way trade-off, and set those parameters for the authentication system.

5.7 Conclusions

We have considered the problem of assigning authentication methods to incoming customer requests that have different probabilities of coming from an impostor. To the best of our knowledge, this is the first attempt at solving this problem using stochastic control theory. We model and solve the problem for two cases: when there are infinitely many servers for each authentication method, and when there is one method with finite capacity and all others have infinite capacity. We do this using a cost-based approach and a constraint-based approach, and obtain structural and

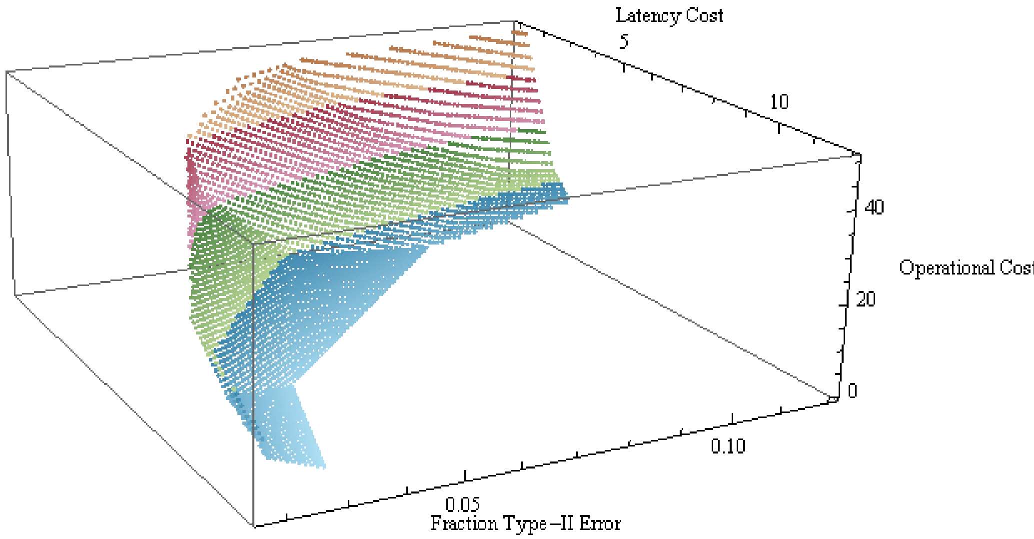

Figure 14: Efficient frontier mapping non-dominated solutions

computational results accordingly.

We conclude that if there are infinitely many servers for each authentication method, then the optimal policy is static. Specifically, under our cost based approach the optimal policy is deterministic and static; and under the constraints based approach the optimal policy is randomized and static.

We also conclude that if one method has finite capacity and exponential service times, then the optimal policy is of trunk reservation form. Namely, under the cost based approach the optimal policy is stationary deterministic trunk reservation; and under the constraint based approach, with positive adjusted rewards, the optimal policy is stationary 2-randomized trunk reservation.

We highlight the fact that the results regarding the structure of the optimal policy are consistent across both approaches. This shows that the results are in a sense robust with respect to the modeling technique. The resulting policy structure is intuitive and easy to implement.

We reiterate that this is the first queuing theoretic work on managing complex authentication systems, and hence, there are many other cases, variations and extensions that can be considered. For example, our ongoing work includes exploring new cases in the constraint-based approach, to include when one method is cheaper and the other is more reliable. In particular, we are working on understanding when the optimal policy has the structure described in Theorem 6, even if one method does not dominate the other in every way. In chapter 6 we go into further detail of future work related to this problem.

CHAPTER VI

CONTRIBUTIONS AND FUTURE WORK

This dissertation focused on developing methodologies for evaluating and optimizing admission and routing control strategies in parallel and tandem queueing systems. The specific systems that we considered were motivated by applications related to traffic routing and user authentication in complex computer networks. In this chapter, we review the main results of our research and connect them to potential future work. First, we summarize the main contributions of each chapter. Then, we highlight questions that remain open and present potential directions for future research, including some proposed strategies to pursue these topics.

6.1 Summary of Contributions

In Chapter 3, we considered an admission control problem for a tandem loss system consisting of two finite capacity service stations. Customers arrive to station 1 according to a Poisson process and a gatekeeper who has complete knowledge of the number of customers at both stations decides to accept or reject each arriving customer. If a customer is rejected, a rejection cost is incurred. If an admitted customer finds that station 2 is full at the time of his service completion at station 1, he leaves the system and a loss cost is incurred. Each station has a single server with exponentially distributed service times. In this chapter we focused on the cases where the buffer at one of the stations is unitary. We also introduced two special policies. The Prudent policy, which only admits an arrival if its probability of being lost is zero; and the Greedy policy, which admits an arrival as long as there is space at the first station. Below, we list our main contributions in this chapter:

- We used uniformization to propose a Markov decision process formulation of this problem.
- We showed that, for $B_1 = 1, B_2 < \infty$, the structure of the optimal policy is threshold and that only the Prudent or Greedy policies can be optimal.
- We provided the exact optimality thresholds for small systems.
- For larger systems we formulated heuristic policies and used extensive numerical experiments to show that these heuristic policies achieve near-optimal performance.
- We also used these numerical experiments to understand the behavior of the stationary probabilities under the Prudent and Greedy policies as $B_2 \to \infty$.

In Chapter 4, we considered the tandem loss system described above, but now we studied the system with identical, arbitrary and finite buffers at both stations. Our main contributions in this chapter are listed below:

- We derived a closed-form expression for the long-run average reward under the Prudent policy.
- We provided a necessary and sufficient condition for the Prudent policy to be optimal.
- We gave a matrix analytic solution to calculate the long-run average reward under the Greedy policy.
- We showed that if the rejection cost is higher than the loss cost, then the Greedy policy is optimal.
- We proved that it is always optimal to admit customers in the states where the Prudent policy admits customers. So, an optimal policy cannot admit customers less frequently than the Prudent policy.

- We gave a full characterization of the optimal policy when the buffer size is two.
- We proposed two heuristic policies and used extensive numerical experiments to show that they perform better than the Prudent and Greedy policies, and in fact, achieve near-optimal performance.

In Chapter 5 we considered a different admission and routing control problem. In particular, we studied an authentication system where requests for authentication arrive from different users. Each request belongs to a class and requests from each class have a fixed probability of coming from an impostor. We assumed requests from each class arrive to the system following independent Poisson processes. A single central controller with full knowledge of the system state must decide how to assign these incoming requests among several available authentication methods. Each authentication method has independent, identically distributed service times. Our goal is to determine how to assign each incoming request to an appropriate authentication method in a way that optimizes the long-run average performance of the system in terms of its operating cost, security, and usability. Below, we list our main contributions in this chapter:

- First, we modeled the problem using a cost-based approach, which assigns a cost to each measure of performance.
  - For the case with several authentication methods, where each method has infinitely many servers we showed that the optimal policy is static and deterministic.
  - For the case where one method has finite capacity and exponential service times, and the rest have infinitely many servers, we showed that the optimal policy is of trunk reservation form, ordered by the costs.
- Then, we modeled the problem using a constraint-based approach, which assumes hard constraints on some of the measures of performance.

  - For the case with several authentication methods, where each method has infinitely many servers we showed that the optimal policy is static and randomized.
  - For the case where one method has finite capacity and exponential service times, and the rest have infinitely many servers, we showed that if the finite-capacity method is better by every measure, then the optimal policy has a 2-randomized trunk reservation form, ordered by an adjusted reward. Also, if an infinite capacity method is better by every measure, then the optimal policy is trivial and routes all requests to that method.

- We illustrated how to use our results to construct an efficient frontier of non-dominated solutions.

In the next section we highlight which questions remain open and potential directions for future work.

6.2 Future Work

Beyond the contributions made in this dissertation, there are still open research questions related to the problems studied here that are worth exploring. These include: proving extensions or generalizations of the results presented; exploring the mathematical properties of the heuristics proposed; and considering related problems that were outside the scope of this dissertation. In this section, we describe some potential avenues of research that we are currently considering.

For the tandem loss system with a unitary buffer presented in Chapter 3, the most direct extension of the results is allowing multiple identical parallel servers at either station. Namely, we believe that Proposition 1 and Theorem 2 can be extended for this case, following the same arguments presented here. More specifically, the sample-path proof of Proposition 1 can be adjusted to allow multiple servers in the second station. Similarly, the proof of Theorem 2 can also be adjusted, using the new stationary distribution of the CTMC with multiple servers at the second station.

Another research opportunity relates to the heuristics proposed in Chapter 3. Kim and Kim [35] proved Conjecture 1 and gave a general expression for the optimality threshold. However, their expression requires many intermediate calculations, and as such is both error-prone and computationally demanding. For this reason, the heuristic policies proposed in Section 3.3.3 are still relevant. While numerical experiments showed good performance of these policies, developing theoretical bounds for their performance is an open research question.

The problem in Chapter 4 is itself a generalization of the problem in Chapter 3, focusing on the case $B_1 = B_2 = B$. As mentioned, many of the results in that chapter hold for any B_1, B_2, however some only hold for $B_1 \geq B_2$. It is of immediate interest, to generalize those results, specifically Proposition 6 and Theorems 3 and 4 to cover the case where $B_1 < B_2$. However, this poses a challenge, as the proofs for the case where $B_1 \geq B_2$, rely heavily on the fact that for a fixed B_2 the set of recurrent states under the Prudent policy is identical for any B_1 such that $B_1 \geq B_2$. Therefore, the proof techniques presented in Chapter 3 do not translate directly for the case where $B_1 < B_2$.

On the other hand, we are currently exploring extensions to Proposition 6 and Theorems 3 and 4 that consider multiple servers at each station, under the assumption that $B_1 \geq B_2$. In this case, the Jackson Network equivalence used to prove Proposition 6 still holds, except with multi-server stations in the network. Similarly, the LP formulation used to prove Theorems 3 and 4 still holds, and follows a similar structure, therefore a similar expression for the reduced costs should be attainable, and then the results for Theorems 3 and 4 would be generalized. Likewise, an extension to the matrix-analytic solution for the long-run average reward under the Greedy policy, to account for multiple servers in the first station should be possible, because the repeating structure of the QBD would be preserved in that situation. However, allowing multiple servers in the second station disrupts the QBD structure. Therefore, a method for calculating the gain under the Greedy policy, with multiple servers at both stations is also an open question.

In Chapter 4 we also have heuristics that have near optimal performance. Therefore, it is worthwhile exploring possible bounds for the heuristics. In fact, in every single test we performed, the PvsG heuristic always performs within a factor of two of the optimal policy, in absolute value. So, we have reason to believe that the gain under the PvsG heuristic, may be bounded below by $-2g^*(B)$.

If we consider the application that motivated the models in Chapters 3 and 4, we find that in the real-life setting these systems are usually more complex than the ones we considered here. Namely, tandem lines often consist of an arbitrary number N of stations. Furthermore, these networks are sometimes configured as parallel-tandem lines where the decision the controller makes includes both admission and routing control. These two problems are of particular interest, as any new insight regarding these systems would be applicable to real applications. However, the results in this dissertation suggest that a purely analytic approach may be intractable. Sheu and Ziedins [63] consider the problem of several parallel tandem lines, where each tandem line has 2 stations and derive asymptotically optimal policies. However, they stress that these policies may perform poorly for systems with few parallel lines. Therefore, alternatives such as heuristics may be considered. One avenue of research we are currently considering is using domain specific knowledge to propose heuristics that perform well, under the regular working conditions of the system.

Now, we move on to the authentication system considered in Chapter 5. Since this is the first queueing theoretic work on managing complex authentication systems, there are many variations and extensions that can be considered. One scenario of practical interest is to extend the results of the constraint-based approach to cases where the limited-capacity server is more secure, but more expensive to use.

Note that the results of Section 5.5.2 assumed that one method was better than the other by all measures. However, this assumption is not necessary for Theorem 6 to hold. That result only requires that $r_i' > 0$ for each class. Therefore, we are currently working on leveraging Theorem 6 under this condition, together with an alternative decision for those classes where $r_i \leq 0$, to achieve a general result that works for any combination of parameters. Specifically, we are interested in a policy that uses the infinite capacity method whenever $r_i \leq 0$ and follows a 2-randomized trunk reservation for classes such that $r_i' > 0$. We suspect such a policy may be optimal in the general case.

The characteristics of actual authentication applications should also be considered. In practice, it is common to have more than one method of authentication with finite capacity. Although each new method that we consider increases the size of the state-space exponentially, the total number of methods available is usually low, so the resulting model may still be computationally tractable. In fact, we believe setting up an LP, similar to $LP3$, should be possible for the case where two methods have finite capacity. However, finding the structure of the optimal policy in that case presents a challenge.

Another assumption that should be re-visited to better approximate the application is the one regarding exponential service times. For the infinite capacity model we have already considered general service times, but not in the finite capacity model. While a Poisson process is a suitable assumption for the arrival process, we need to consider other service time distributions. However, without the exponential service times assumption the discretization procedure in Section 5.5.1 would fail. On the other hand, Fan-Orzechowski and Feinberg [18] model a similar parallel queueing system as a semi-Markov process with exponential service time. Their model could be extended to account for non-exponential service times.

Another model of interest in the authentication setting is the classification tool. We assumed that the classification of user-requests into I classes based on impostor probabilities was done *a priori*. A model which takes each request's underlying characteristics, such as location, request type, device, etc. into account and returns an appropriate class is of practical interest. Techniques such as statistical clustering or machine learning are possibly the most appropriate for this purpose. Such a model could be implemented jointly with the models proposed here to produce an integrated authentication system.

In this dissertation we have used queueing theory and Markov decision processes to study and solve two admission and routing control problems in parallel and tandem computer networks. However, there are many more applications in computer networks, including routing data packets, access to secure systems and data, capacity and resource allocation, task scheduling, etc. where similar techniques can be applied. We hope that this thesis serves as an initial step for us and other researchers in the field to apply these tools to many potential applications arising in computer networks.

APPENDIX A

SUPPLEMENTAL MATERIAL FOR CHAPTER 3

In this section we provide the full expressions for the terms defined in Chapter 3: $c_*(B_2)$, $f_1(\lambda, \mu_1, \mu_2, B_2)$ and $f_2(\lambda, \mu_1, \mu_2, B_2)$ for each $B_2 \leq 10$.

A.1 Threshold values for Case 2

Recall from section 3.3.1 that for each B_2 we have a expressions for the critical cost $c_*(B_2)$ of the form:

$$c_*(B_2) = \left(1 + \frac{\alpha(B_2)}{\beta(B_2)}\right)$$

where

$$\alpha(B_2) = \mu_2^{B_2+1} \sum_{i=1}^{\lceil B_2/2 \rceil} \Big[\big(\lambda^{B_2-i} + \mathbf{1}_{\{(B_2-i)>(i-1)\}} \lambda^{i-1} (\mu_1 + \mu_2)^{B_2-2i+1} \big) \sum_{j=0}^{i-1} \left(\mu_1^{i-j-1} \mu_2^j \binom{i-1}{j} \frac{1}{j!} \prod_{k=1}^{j} (B_2 - i + k) \right) \Big].$$

We have that $\beta(B_2)$ is given by

$$\beta(1) = \lambda\mu_1 + \lambda\mu_2 + \mu_1\mu_2$$

$$\beta(2) = \mu_1\mu_2^2(\mu_1 + \mu_2) + \lambda\mu_2(\mu_1 + \mu_2)^2 + \lambda^2 \left(\mu_1^2 + \mu_1\mu_2 + \mu_2^2\right)$$

$$\beta(3) = \mu_1\mu_2^3(\mu_1 + \mu_2)^2 + \lambda^3 \left(\mu_1^3 + \mu_1^2\mu_2 + \mu_1\mu_2^2 + \mu_2^3\right) + \lambda\mu_2^2 \left(\mu_1^3 + 3\mu_1^2\mu_2 + 4\mu_1\mu_2^2 + \mu_2^3\right) + \lambda^2\mu_2 \left(\mu_1^3 + 2\mu_1^2\mu_2 + 3\mu_1\mu_2^2 + 2\mu_2^3\right)$$

$$\beta(4) = \mu_1\mu_2^4(\mu_1 + \mu_2)^3 + \lambda^4 \left(\mu_1^4 + \mu_1^3\mu_2 + \mu_1^2\mu_2^2 + \mu_1\mu_2^3 + \mu_2^4\right) + \lambda\mu_2^3 \left(\mu_1^4 + 4\mu_1^3\mu_2 + 8\mu_1^2\mu_2^2 + 6\mu_1\mu_2^3 + \mu_2^4\right) + \lambda^3\mu_2 \left(\mu_1^4 + 2\mu_1^3\mu_2 + 3\mu_1^2\mu_2^2 + 4\mu_1\mu_2^3 + 3\mu_2^4\right) + \lambda^2\mu_2^2 \left(\mu_1^4 + 3\mu_1^3\mu_2 + 6\mu_1^2\mu_2^2 + 8\mu_1\mu_2^3 + 3\mu_2^4\right)$$

$$
\begin{aligned}
\beta(5) &= \mu_1\mu_2^5(\mu_1+\mu_2)^4 + \lambda\mu_2^4(\mu_1+\mu_2)^2\left(\mu_1^3+3\mu_1^2\mu_2+6\mu_1\mu_2^2+\mu_2^3\right)\\
&+ \lambda^2\mu_2^3\left(\mu_1^5+4\mu_1^4\mu_2+10\mu_1^3\mu_2^2+18\mu_1^2\mu_2^3+16\mu_1\mu_2^4+4\mu_2^5\right)\\
&+ \lambda^3\mu_2^2\left(\mu_1^5+3\mu_1^4\mu_2+6\mu_1^3\mu_2^2+10\mu_1^2\mu_2^3+13\mu_1\mu_2^4+6\mu_2^5\right)\\
&+ \lambda^4\mu_2\left(\mu_1^5+2\mu_1^4\mu_2+3\mu_1^3\mu_2^2+4\mu_1^2\mu_2^3+5\mu_1\mu_2^4+4\mu_2^5\right)\\
&+ \lambda^5\left(\mu_1^5+\mu_1^4\mu_2+\mu_1^3\mu_2^2+\mu_1^2\mu_2^3+\mu_1\mu_2^4+\mu_2^5\right)\\
\beta(6) &= \mu_1\mu_2^6(\mu_1+\mu_2)^5 + \lambda\mu_2^5(\mu_1+\mu_2)^3\left(\mu_1^3+3\mu_1^2\mu_2+7\mu_1\mu_2^2+\mu_2^3\right)\\
&+ \lambda^2\mu_2^4\left(\mu_1^6+5\mu_1^5\mu_2+15\mu_1^4\mu_2^2+33\mu_1^3\mu_2^3+44\mu_1^2\mu_2^4+27\mu_1\mu_2^5+5\mu_2^6\right)\\
&+ \lambda^3\mu_2^3\left(\mu_1^6+4\mu_1^5\mu_2+10\mu_1^4\mu_2^2+20\mu_1^3\mu_2^3+33\mu_1^2\mu_2^4+32\mu_1\mu_2^5+10\mu_2^6\right)\\
&+ \lambda^4\mu_2^2\left(\mu_1^6+3\mu_1^5\mu_2+6\mu_1^4\mu_2^2+10\mu_1^3\mu_2^3+15\mu_1^2\mu_2^4+19\mu_1\mu_2^5+10\mu_2^6\right)\\
&+ \lambda^5\mu_2\left(\mu_1^6+2\mu_1^5\mu_2+3\mu_1^4\mu_2^2+4\mu_1^3\mu_2^3+5\mu_1^2\mu_2^4+6\mu_1\mu_2^5+5\mu_2^6\right)\\
&+ \lambda^6\left(\mu_1^6+\mu_1^5\mu_2+\mu_1^4\mu_2^2+\mu_1^3\mu_2^3+\mu_1^2\mu_2^4+\mu_1\mu_2^5+\mu_2^6\right)\\
\beta(7) &= \mu_1\mu_2^7(\mu_1+\mu_2)^6 + \lambda\mu_2^6(\mu_1+\mu_2)^4\left(\mu_1^3+3\mu_1^2\mu_2+8\mu_1\mu_2^2+\mu_2^3\right)\\
&+ \lambda^2\mu_2^5(\mu_1+\mu_2)^2\left(\mu_1^5+4\mu_1^4\mu_2+12\mu_1^3\mu_2^2+26\mu_1^2\mu_2^3+29\mu_1\mu_2^4+6\mu_2^5\right)\\
&+ \lambda^3\mu_2^4\left(\mu_1^7+5\mu_1^6\mu_2+15\mu_1^5\mu_2^2+35\mu_1^4\mu_2^3+68\mu_1^3\mu_2^4+93\mu_1^2\mu_2^5+65\mu_1\mu_2^6+15\mu_2^7\right)\\
&+ \lambda^4\mu_2^3\left(\mu_1^7+4\mu_1^6\mu_2+10\mu_1^5\mu_2^2+20\mu_1^4\mu_2^3+35\mu_1^3\mu_2^4+54\mu_1^2\mu_2^5+55\mu_1\mu_2^6+20\mu_2^7\right)\\
&+ \lambda^5\mu_2^2\left(\mu_1^7+3\mu_1^6\mu_2+6\mu_1^5\mu_2^2+10\mu_1^4\mu_2^3+15\mu_1^3\mu_2^4+21\mu_1^2\mu_2^5+26\mu_1\mu_2^6+15\mu_2^7\right)\\
&+ \lambda^6\mu_2\left(\mu_1^7+2\mu_1^6\mu_2+3\mu_1^5\mu_2^2+4\mu_1^4\mu_2^3+5\mu_1^3\mu_2^4+6\mu_1^2\mu_2^5+7\mu_1\mu_2^6+6\mu_2^7\right)\\
&+ \lambda^7\left(\mu_1^7+\mu_1^6\mu_2+\mu_1^5\mu_2^2+\mu_1^4\mu_2^3+\mu_1^3\mu_2^4+\mu_1^2\mu_2^5+\mu_1\mu_2^6+\mu_2^7\right)
\end{aligned}
$$

$$
\begin{aligned}
\beta(8) &= \mu_1\mu_2^8(\mu_1+\mu_2)^7 + \lambda\mu_2^7(\mu_1+\mu_2)^5\left(\mu_1^3+3\mu_1^2\mu_2+9\mu_1\mu_2^2+\mu_2^3\right)\\
&+\lambda^2\mu_2^6(\mu_1+\mu_2)^3\left(\mu_1^5+4\mu_1^4\mu_2+13\mu_1^3\mu_2^2+30\mu_1^2\mu_2^3+37\mu_1\mu_2^4+7\mu_2^5\right)\\
&+\lambda^3\mu_2^5\left(\mu_1^8+6\mu_1^7\mu_2+21\mu_1^6\mu_2^2+56\mu_1^5\mu_2^3+124\mu_1^4\mu_2^4+210\mu_1^3\mu_2^5+221\mu_1^2\mu_2^6\right.\\
&\left.+116\mu_1\mu_2^7+21\mu_2^8\right)+\lambda^4\mu_2^4\left(\mu_1^8+5\mu_1^7\mu_2+15\mu_1^6\mu_2^2+35\mu_1^5\mu_2^3+70\mu_1^4\mu_2^4\right.\\
&\left.+124\mu_1^3\mu_2^5+170\mu_1^2\mu_2^6+130\mu_1\mu_2^7+35\mu_2^8\right)+\lambda^5\mu_2^3\left(\mu_1^8+4\mu_1^7\mu_2+10\mu_1^6\mu_2^2\right.\\
&\left.+20\mu_1^5\mu_2^3+35\mu_1^4\mu_2^4+56\mu_1^3\mu_2^5+82\mu_1^2\mu_2^6+86\mu_1\mu_2^7+35\mu_2^8\right)+\lambda^6\mu_2^2\left(\mu_1^8\right.\\
&\left.+3\mu_1^7\mu_2+6\mu_1^6\mu_2^2+10\mu_1^5\mu_2^3+15\mu_1^4\mu_2^4+21\mu_1^3\mu_2^5+28\mu_1^2\mu_2^6+34\mu_1\mu_2^7+21\mu_2^8\right)\\
&+\lambda^7\mu_2\left(\mu_1^8+2\mu_1^7\mu_2+3\mu_1^6\mu_2^2+4\mu_1^5\mu_2^3+5\mu_1^4\mu_2^4+6\mu_1^3\mu_2^5+7\mu_1^2\mu_2^6\right.\\
&\left.+8\mu_1\mu_2^7+7\mu_2^8\right)+\lambda^8\left(\mu_1^8+\mu_1^7\mu_2+\mu_1^6\mu_2^2+\mu_1^5\mu_2^3+\mu_1^4\mu_2^4+\mu_1^3\mu_2^5+\mu_1^2\mu_2^6+\mu_1\mu_2^7+\mu_2^8\right)\\
\beta(9) &= \mu_1\mu_2^9(\mu_1+\mu_2)^8+\lambda\mu_2^8(\mu_1+\mu_2)^6\left(\mu_1^3+3\mu_1^2\mu_2+10\mu_1\mu_2^2+\mu_2^3\right)\\
&+\lambda^2\mu_2^7(\mu_1+\mu_2)^4\left(\mu_1^5+4\mu_1^4\mu_2+14\mu_1^3\mu_2^2+34\mu_1^2\mu_2^3+46\mu_1\mu_2^4+8\mu_2^5\right)\\
&+\lambda^3\mu_2^6(\mu_1+\mu_2)^2\left(\mu_1^7+5\mu_1^6\mu_2+17\mu_1^5\mu_2^2+45\mu_1^4\mu_2^3+101\mu_1^3\mu_2^4+164\mu_1^2\mu_2^5+133\mu_1\mu_2^6\right.\\
&\left.+28\mu_2^7\right)+\lambda^9\left(\mu_1^9+\mu_1^8\mu_2+\mu_1^7\mu_2^2+\mu_1^6\mu_2^3+\mu_1^5\mu_2^4+\mu_1^4\mu_2^5+\mu_1^3\mu_2^6+\mu_1^2\mu_2^7+\mu_1\mu_2^8+\mu_2^9\right)\\
&+\lambda^8\mu_2\left(\mu_1^9+2\mu_1^8\mu_2+3\mu_1^7\mu_2^2+4\mu_1^6\mu_2^3+5\mu_1^5\mu_2^4+6\mu_1^4\mu_2^5+7\mu_1^3\mu_2^6+8\mu_1^2\mu_2^7\right.\\
&\left.+9\mu_1\mu_2^8+8\mu_2^9\right)+\lambda^7\mu_2^2\left(\mu_1^9+3\mu_1^8\mu_2+6\mu_1^7\mu_2^2+10\mu_1^6\mu_2^3+15\mu_1^5\mu_2^4+21\mu_1^4\mu_2^5+28\mu_1^3\mu_2^6\right.\\
&\left.+36\mu_1^2\mu_2^7+43\mu_1\mu_2^8+28\mu_2^9\right)+\lambda^6\mu_2^3\left(\mu_1^9+4\mu_1^8\mu_2+10\mu_1^7\mu_2^2+20\mu_1^6\mu_2^3+35\mu_1^5\mu_2^4\right.\\
&\left.+56\mu_1^4\mu_2^5+84\mu_1^3\mu_2^6+118\mu_1^2\mu_2^7+126\mu_1\mu_2^8+56\mu_2^9\right)+\lambda^4\mu_2^5\left(\mu_1^9+6\mu_1^8\mu_2+21\mu_1^7\mu_2^2\right.\\
&\left.+56\mu_1^6\mu_2^3+126\mu_1^5\mu_2^4+250\mu_1^4\mu_2^5+411\mu_1^3\mu_2^6+456\mu_1^2\mu_2^7+266\mu_1\mu_2^8+56\mu_2^9\right)\\
&+\lambda^5\mu_2^4\left(\mu_1^9+5\mu_1^8\mu_2+15\mu_1^7\mu_2^2+35\mu_1^6\mu_2^3+70\mu_1^5\mu_2^4+126\mu_1^4\mu_2^5\right.\\
&\left.+208\mu_1^3\mu_2^6+283\mu_1^2\mu_2^7+231\mu_1\mu_2^8+70\mu_2^9\right)
\end{aligned}
$$

$$\beta(10) = \mu_1\mu_2^{10}(\mu_1+\mu_2)^9 + \lambda\mu_2^9(\mu_1+\mu_2)^7\left(\mu_1^3+3\mu_1^2\mu_2+11\mu_1\mu_2^2+\mu_2^3\right)$$
$$+\lambda^2\mu_2^8(\mu_1+\mu_2)^5\left(\mu_1^5+4\mu_1^4\mu_2+15\mu_1^3\mu_2^2+38\mu_1^2\mu_2^3+56\mu_1\mu_2^4+9\mu_2^5\right)$$
$$+\lambda^3\mu_2^7(\mu_1+\mu_2)^3\left(\mu_1^7+5\mu_1^6\mu_2+18\mu_1^5\mu_2^2+50\mu_1^4\mu_2^3+119\mu_1^3\mu_2^4+207\mu_1^2\mu_2^5\right.$$
$$\left.+180\mu_1\mu_2^6+36\mu_2^7\right)+\lambda^{10}\left(\mu_1^{10}+\mu_1^9\mu_2+\mu_1^8\mu_2^2+\mu_1^7\mu_2^3+\mu_1^6\mu_2^4+\mu_1^5\mu_2^5+\mu_1^4\mu_2^6\right.$$
$$\left.+\mu_1^3\mu_2^7+\mu_1^2\mu_2^8+\mu_1\mu_2^9+\mu_2^{10}\right)+\lambda^9\mu_2\left(\mu_1^{10}+2\mu_1^9\mu_2+3\mu_1^8\mu_2^2+4\mu_1^7\mu_2^3+5\mu_1^6\mu_2^4\right.$$
$$\left.+6\mu_1^5\mu_2^5+7\mu_1^4\mu_2^6+8\mu_1^3\mu_2^7+9\mu_1^2\mu_2^8+10\mu_1\mu_2^9+9\mu_2^{10}\right)+\lambda^8\mu_2^2\left(\mu_1^{10}+3\mu_1^9\mu_2+6\mu_1^8\mu_2^2\right.$$
$$\left.+10\mu_1^7\mu_2^3+15\mu_1^6\mu_2^4+21\mu_1^5\mu_2^5+28\mu_1^4\mu_2^6+36\mu_1^3\mu_2^7+45\mu_1^2\mu_2^8+53\mu_1\mu_2^9+36\mu_2^{10}\right)$$
$$+\lambda^7\mu_2^3\left(\mu_1^{10}+4\mu_1^9\mu_2+10\mu_1^8\mu_2^2+20\mu_1^7\mu_2^3+35\mu_1^6\mu_2^4+56\mu_1^5\mu_2^5+84\mu_1^4\mu_2^6+120\mu_1^3\mu_2^7\right.$$
$$\left.+163\mu_1^2\mu_2^8+176\mu_1\mu_2^9+84\mu_2^{10}\right)+\lambda^4\mu_2^6\left(\mu_1^{10}+7\mu_1^9\mu_2+28\mu_1^8\mu_2^2+84\mu_1^7\mu_2^3\right.$$
$$\left.+210\mu_1^6\mu_2^4+460\mu_1^5\mu_2^5+862\mu_1^4\mu_2^6+1208\mu_1^3\mu_2^7+1064\mu_1^2\mu_2^8+490\mu_1\mu_2^9+84\mu_2^{10}\right)$$
$$+\lambda^6\mu_2^4\left(\mu_1^{10}+5\mu_1^9\mu_2+15\mu_1^8\mu_2^2+35\mu_1^7\mu_2^3+70\mu_1^6\mu_2^4+126\mu_1^5\mu_2^5+210\mu_1^4\mu_2^6\right.$$
$$\left.+328\mu_1^3\mu_2^7+441\mu_1^2\mu_2^8+378\mu_1\mu_2^9+126\mu_2^{10}\right)$$
$$+\lambda^5\mu_2^5\left(\mu_1^{10}+6\mu_1^9\mu_2+21\mu_1^8\mu_2^2+56\mu_1^7\mu_2^3+126\mu_1^6\mu_2^4+252\mu_1^5\mu_2^5+460\mu_1^4\mu_2^6\right.$$
$$\left.+732\mu_1^3\mu_2^7+840\mu_1^2\mu_2^8+532\mu_1\mu_2^9+126\mu_2^{10}\right)$$

A.2 Values of f_1 expressions in Proposition 2

In this section, we provide the f_1 expressions used in the proof of Proposition 2 for $B_2 = 1, ..., 10$.

$$f_1(\lambda,\mu_1,\mu_2,1)=0$$
$$f_1(\lambda,\mu_1,\mu_2,2)=\lambda^2\mu_2^2+2\lambda\mu_1\mu_2^2+\mu_1^2\mu_2^2+\lambda\mu_2^3+\mu_1\mu_2^3$$
$$f_1(\lambda,\mu_1,\mu_2,3)=\lambda^3\mu_1\mu_2^2+2\lambda^2\mu_1^2\mu_2^2+\lambda\mu_1^3\mu_2^2+\lambda^3\mu_2^3+3\lambda^2\mu_1\mu_2^3+3\lambda\mu_1^2\mu_2^3+\mu_1^3\mu_2^3$$
$$+2\lambda^2\mu_2^4+4\lambda\mu_1\mu_2^4+2\mu_1^2\mu_2^4+\lambda\mu_2^5+\mu_1\mu_2^5$$
$$f_1(\lambda,\mu_1,\mu_2,4)=(\lambda+\mu_1)\mu_2^2(\lambda^2\mu_1^2(\lambda+\mu_1)+\lambda\mu_1(\lambda+\mu_1)^2\mu_2+(\lambda+\mu_1)^3\mu_2^2$$
$$+(3\lambda^2+5\lambda\mu_1+3\mu_1^2)\mu_2^3+3(\lambda+\mu_1)\mu_2^4+\mu_2^5)$$

$$f_1(\lambda,\mu_1,\mu_2,5) = (\lambda+\mu_1)\mu_2^2(\lambda^3\mu_1^3(\lambda+\mu_1) + \lambda^2\mu_1^2(\lambda+\mu_1)^2\mu_2 + \lambda\mu_1(\lambda+\mu_1)^3\mu_2^2$$
$$+ (\lambda+\mu_1)^4\mu_2^3 + (\lambda+\mu_1)(4\lambda^2+5\lambda\mu_1+4\mu_1^2)\mu_2^4 + 2(3\lambda^2+5\lambda\mu_1+3\mu_1^2)\mu_2^5$$
$$+ 4(\lambda+\mu_1)\mu_2^6 + \mu_2^7)$$
$$f_1(\lambda,\mu_1,\mu_2,6) = (\lambda+\mu_1)\mu_2^2(\lambda^4\mu_1^4(\lambda+\mu_1) + \lambda^3\mu_1^3(\lambda+\mu_1)^2\mu_2 + \lambda^2\mu_1^2(\lambda+\mu_1)^3\mu_2^2$$
$$+ \lambda\mu_1(\lambda+\mu_1)^4\mu_2^3 + (\lambda+\mu_1)^5\mu_2^4 + (\lambda^2+\lambda\mu_1+\mu_1^2)(5\lambda^2+9\lambda\mu_1+5\mu_1^2)\mu_2^5$$
$$+ 2(\lambda+\mu_1)(5\lambda^2+6\lambda\mu_1+5\mu_1^2)\mu_2^6 + (10\lambda^2+17\lambda\mu_1+10\mu_1^2)\mu_2^7 + 5(\lambda+\mu_1)\mu_2^8 + \mu_2^9)$$
$$f_1(\lambda,\mu_1,\mu_2,7) = (\lambda+\mu_1)\mu_2^2(\lambda^5\mu_1^5(\lambda+\mu_1) + \lambda^4\mu_1^4(\lambda+\mu_1)^2\mu_2 + \lambda^3\mu_1^3(\lambda+\mu_1)^3\mu_2^2$$
$$+ \lambda^2\mu_1^2(\lambda+\mu_1)^4\mu_2^3 + \lambda\mu_1(\lambda+\mu_1)^5\mu_2^4 + (\lambda+\mu_1)^6\mu_2^5$$
$$+ 2(\lambda+\mu_1)(\lambda^2+\lambda\mu_1+\mu_1^2)(3\lambda^2+4\lambda\mu_1+3\mu_1^2)\mu_2^6$$
$$+ (15\lambda^4+40\lambda^3\mu_1+53\lambda^2\mu_1^2+40\lambda\mu_1^3+15\mu_1^4)\mu_2^7$$
$$+ 5(\lambda+\mu_1)(4\lambda^2+5\lambda\mu_1+4\mu_1^2)\mu_2^8 + (15\lambda^2+26\lambda\mu_1+15\mu_1^2)\mu_2^9$$
$$+ 6(\lambda+\mu_1)\mu_2^{10} + \mu_2^{11})$$
$$f_1(\lambda,\mu_1,\mu_2,8) = (\lambda+\mu_1)\mu_2^2(\lambda^6\mu_1^6(\lambda+\mu_1) + \lambda^5\mu_1^5(\lambda+\mu_1)^2\mu_2 + \lambda^4\mu_1^4(\lambda+\mu_1)^3\mu_2^2$$
$$+ \lambda^3\mu_1^3(\lambda+\mu_1)^4\mu_2^3 + \lambda^2\mu_1^2(\lambda+\mu_1)^5\mu_2^4 + \lambda\mu_1(\lambda+\mu_1)^6\mu_2^5 + (\lambda+\mu_1)^7\mu_2^6$$
$$+ (7\lambda^6+27\lambda^5\mu_1+55\lambda^4\mu_1^2+69\lambda^3\mu_1^3+55\lambda^2\mu_1^4+27\lambda\mu_1^5+7\mu_1^6)\mu_2^7$$
$$+ (\lambda+\mu_1)(21\lambda^4+44\lambda^3\mu_1+61\lambda^2\mu_1^2+44\lambda\mu_1^3+21\mu_1^4)\mu_2^8$$
$$+ (35\lambda^4+95\lambda^3\mu_1+126\lambda^2\mu_1^2+95\lambda\mu_1^3+35\mu_1^4)\mu_2^9$$
$$+ (\lambda+\mu_1)(35\lambda^2+46\lambda\mu_1+35\mu_1^2)\mu_2^{10}$$
$$+ (21\lambda^2+37\lambda\mu_1+21\mu_1^2)\mu_2^{11} + 7(\lambda+\mu_1)\mu_2^{12} + \mu_2^{13})$$

$$f_1(\lambda,\mu_1,\mu_2,9) = (\lambda+\mu_1)\mu_2^2(\lambda^7\mu_1^7(\lambda+\mu_1) + \lambda^6\mu_1^6(\lambda+\mu_1)^2\mu_2 + \lambda^5\mu_1^5(\lambda+\mu_1)^3\mu_2^2$$
$$+ \lambda^4\mu_1^4(\lambda+\mu_1)^4\mu_2^3 + \lambda^3\mu_1^3(\lambda+\mu_1)^5\mu_2^4 + \lambda^2\mu_1^2(\lambda+\mu_1)^6\mu_2^5 + \lambda\mu_1(\lambda+\mu_1)^7\mu_2^6$$
$$+ (\lambda+\mu_1)^8\mu_2^7 + (\lambda+\mu_1)(8\lambda^6 + 27\lambda^5\mu_1 + 56\lambda^4\mu_1^2 + 69\lambda^3\mu_1^3 + 56\lambda^2\mu_1^4 + 27\lambda\mu_1^5$$
$$+ 8\mu_1^6)\mu_2^8 + (28\lambda^6 + 98\lambda^5\mu_1 + 185\lambda^4\mu_1^2 + 226\lambda^3\mu_1^3 + 185\lambda^2\mu_1^4 + 98\lambda\mu_1^5 + 28\mu_1^6)\mu_2^9$$
$$+ (\lambda+\mu_1)(56\lambda^4 + 119\lambda^3\mu_1 + 162\lambda^2\mu_1^2 + 119\lambda\mu_1^3 + 56\mu_1^4)\mu_2^{10}$$
$$+ 2(35\lambda^4 + 98\lambda^3\mu_1 + 131\lambda^2\mu_1^2 + 98\lambda\mu_1^3 + 35\mu_1^4)\mu_2^{11}$$
$$+ 7(\lambda+\mu_1)(8\lambda^2 + 11\lambda\mu_1 + 8\mu_1^2)\mu_2^{12} + 2(14\lambda^2 + 25\lambda\mu_1 + 14\mu_1^2)\mu_2^{13}$$
$$+ 8(\lambda+\mu_1)\mu_2^{14} + \mu_2^{15})$$

$$f_1(\lambda,\mu_1,\mu_2,10) = (\lambda+\mu_1)\mu_2^2(\lambda^8\mu_1^8(\lambda+\mu_1) + \lambda^7\mu_1^7(\lambda+\mu_1)^2\mu_2 + \lambda^6\mu_1^6(\lambda+\mu_1)^3\mu_2^2$$
$$+ \lambda^5\mu_1^5(\lambda+\mu_1)^4\mu_2^3 + \lambda^4\mu_1^4(\lambda+\mu_1)^5\mu_2^4 + \lambda^3\mu_1^3(\lambda+\mu_1)^6\mu_2^5 + \lambda^2\mu_1^2(\lambda+\mu_1)^7\mu_2^6$$
$$+ \lambda\mu_1(\lambda+\mu_1)^8\mu_2^7 + (\lambda+\mu_1)^9\mu_2^8 + (9\lambda^8 + 44\lambda^7\mu_1 + 119\lambda^6\mu_1^2 + 209\lambda^5\mu_1^3$$
$$+ 251\lambda^4\mu_1^4 + 209\lambda^3\mu_1^5 + 119\lambda^2\mu_1^6 + 44\lambda\mu_1^7 + 9\mu_1^8)\mu_2^9$$
$$+ (\lambda+\mu_1)(36\lambda^6 + 104\lambda^5\mu_1 + 197\lambda^4\mu_1^2 + 234\lambda^3\mu_1^3 + 197\lambda^2\mu_1^4 + 104\lambda\mu_1^5 + 36\mu_1^6)\mu_2^{10}$$
$$+ 2(42\lambda^6 + 147\lambda^5\mu_1 + 273\lambda^4\mu_1^2 + 331\lambda^3\mu_1^3 + 273\lambda^2\mu_1^4 + 147\lambda\mu_1^5 + 42\mu_1^6)\mu_2^{11}$$
$$+ 14(\lambda+\mu_1)(9\lambda^4 + 20\lambda^3\mu_1 + 27\lambda^2\mu_1^2 + 20\lambda\mu_1^3 + 9\mu_1^4)\mu_2^{12}$$
$$+ (126\lambda^4 + 364\lambda^3\mu_1 + 491\lambda^2\mu_1^2 + 364\lambda\mu_1^3 + 126\mu_1^4)\mu_2^{13} + 12(\lambda+\mu_1)(7\lambda^2 + 10\lambda\mu_1$$
$$+ 7\mu_1^2)\mu_2^{14} + (36\lambda^2 + 65\lambda\mu_1 + 36\mu_1^2)\mu_2^{15} + 9(\lambda+\mu_1)\mu_2^{16} + \mu_2^{17}).$$

A.3 Values of f_2 expressions in Proposition 3

In this section, we provide the f_2 expressions used in the proof of Propositions 3 and 4 for $B_2 = 1, ..., 10$.

$$f_2(\lambda,\mu_1,\mu_2,1) = 0$$

$$f_2(\lambda,\mu_1,\mu_2,2) = \lambda^3\mu_2^2 + 2\lambda^2\mu_1\mu_2^2 + 2\lambda\mu_1^2\mu_2^2 + \mu_1^3\mu_2^2 + 2\lambda^2\mu_2^3 + 4\lambda\mu_1\mu_2^3 + 2\mu_1^2\mu_2^3$$
$$+ \lambda\mu_2^4 + \mu_1\mu_2^4$$

$$f_2(\lambda,\mu_1,\mu_2,3) = \lambda^4\mu_1\mu_2^2 + 2\lambda^3\mu_1^2\mu_2^2 + 2\lambda^2\mu_1^3\mu_2^2 + \lambda\mu_1^4\mu_2^2 + \lambda^4\mu_2^3 + 4\lambda^3\mu_1\mu_2^3$$
$$+ 6\lambda^2\mu_1^2\mu_2^3 + 4\lambda\mu_1^3\mu_2^3 + \mu_1^4\mu_2^3 + 3\lambda^3\mu_2^4 + 8\lambda^2\mu_1\mu_2^4 + 8\lambda\mu_1^2\mu_2^4 + 3\mu_1^3\mu_2^4 + 3\lambda^2\mu_2^5$$
$$+ 6\lambda\mu_1\mu_2^5 + 3\mu_1^2\mu_2^5 + \lambda\mu_2^6 + \mu_1\mu_2^6$$
$$f_2(\lambda,\mu_1,\mu_2,4) = (\lambda+\mu_1)\mu_2^2(\lambda^2\mu_1^2(\lambda^2+\lambda\mu_1+\mu_1^2) + \lambda\mu_1(\lambda+\mu_1)^3\mu_2 + (\lambda+\mu_1)^4\mu_2^2$$
$$+ (\lambda+\mu_1)(4\lambda^2+5\lambda\mu_1+4\mu_1^2)\mu_2^3 + 2(3\lambda^2+5\lambda\mu_1+3\mu_1^2)\mu_2^4 + 4(\lambda+\mu_1)\mu_2^5 + \mu_2^6)$$
$$f_2(\lambda,\mu_1,\mu_2,5) = (\lambda+\mu_1)\mu_2^2(\lambda^3\mu_1^3(\lambda^2+\lambda\mu_1+\mu_1^2) + \lambda^2\mu_1^2(\lambda+\mu_1)^3\mu_2$$
$$+ \lambda\mu_1(\lambda+\mu_1)^4\mu_2^2 + (\lambda+\mu_1)^5\mu_2^3 + (\lambda^2+\lambda\mu_1+\mu_1^2)(5\lambda^2+9\lambda\mu_1+5\mu_1^2)\mu_2^4$$
$$+ 2(\lambda+\mu_1)(5\lambda^2+6\lambda\mu_1+5\mu_1^2)\mu_2^5 + (10\lambda^2+17\lambda\mu_1+10\mu_1^2)\mu_2^6 + 5(\lambda+\mu_1)\mu_2^7 + \mu_2^8)$$
$$f_2(\lambda,\mu_1,\mu_2,6) = (\lambda+\mu_1)\mu_2^2(\lambda^4\mu_1^4(\lambda^2+\lambda\mu_1+\mu_1^2) + \lambda^3\mu_1^3(\lambda+\mu_1)^3\mu_2 + \lambda^2\mu_1^2(\lambda$$
$$+ \mu_1)^4\mu_2^2 + \lambda\mu_1(\lambda+\mu_1)^5\mu_2^3 + (\lambda+\mu_1)^6\mu_2^4 + 2(\lambda+\mu_1)(\lambda^2+\lambda\mu_1+\mu_1^2)(3\lambda^2+4\lambda\mu_1$$
$$+ 3\mu_1^2)\mu_2^5 + (15\lambda^4+40\lambda^3\mu_1+53\lambda^2\mu_1^2+40\lambda\mu_1^3+15\mu_1^4)\mu_2^6 + 5(\lambda+\mu_1)(4\lambda^2+5\lambda\mu_1$$
$$+ 4\mu_1^2)\mu_2^7 + (15\lambda^2+26\lambda\mu_1+15\mu_1^2)\mu_2^8 + 6(\lambda+\mu_1)\mu_2^9 + \mu_2^{10})$$
$$f_2(\lambda,\mu_1,\mu_2,7) = (\lambda+\mu_1)\mu_2^2(\lambda^5\mu_1^5(\lambda^2+\lambda\mu_1+\mu_1^2) + \lambda^4\mu_1^4(\lambda+\mu_1)^3\mu_2 + \lambda^3\mu_1^3(\lambda$$
$$+ \mu_1)^4\mu_2^2 + \lambda^2\mu_1^2(\lambda+\mu_1)^5\mu_2^3 + \lambda\mu_1(\lambda+\mu_1)^6\mu_2^4 + (\lambda+\mu_1)^7\mu_2^5 + (7\lambda^6+27\lambda^5\mu_1$$
$$+ 55\lambda^4\mu_1^2 + 69\lambda^3\mu_1^3 + 55\lambda^2\mu_1^4 + 27\lambda\mu_1^5 + 7\mu_1^6)\mu_2^6 + (\lambda+\mu_1)(21\lambda^4+44\lambda^3\mu_1$$
$$+ 61\lambda^2\mu_1^2 + 44\lambda\mu_1^3 + 21\mu_1^4)\mu_2^7 + (35\lambda^4+95\lambda^3\mu_1+126\lambda^2\mu_1^2+95\lambda\mu_1^3+35\mu_1^4)\mu_2^8$$
$$+ (\lambda+\mu_1)(35\lambda^2+46\lambda\mu_1+35\mu_1^2)\mu_2^9 + (21\lambda^2+37\lambda\mu_1+21\mu_1^2)\mu_2^{10}$$
$$+ 7(\lambda+\mu_1)\mu_2^{11} + \mu_2^{12})$$

$$
\begin{aligned}
f_2(\lambda,\mu_1,\mu_2,8) &= (\lambda+\mu_1)\mu_2^2(\lambda^6\mu_1^6(\lambda^2+\lambda\mu_1+\mu_1^2)+\lambda^5\mu_1^5(\lambda+\mu_1)^3\mu_2+\lambda^4\mu_1^4(\lambda \\
&\quad +\mu_1)^4\mu_2^2+\lambda^3\mu_1^3(\lambda+\mu_1)^5\mu_2^3+\lambda^2\mu_1^2(\lambda+\mu_1)^6\mu_2^4+\lambda\mu_1(\lambda+\mu_1)^7\mu_2^5+(\lambda+\mu_1)^8\mu_2^6 \\
&\quad +(\lambda+\mu_1)(8\lambda^6+27\lambda^5\mu_1+56\lambda^4\mu_1^2+69\lambda^3\mu_1^3+56\lambda^2\mu_1^4+27\lambda\mu_1^5+8\mu_1^6)\mu_2^7+(28\lambda^6 \\
&\quad +98\lambda^5\mu_1+185\lambda^4\mu_1^2+226\lambda^3\mu_1^3+185\lambda^2\mu_1^4+98\lambda\mu_1^5+28\mu_1^6)\mu_2^8+(\lambda+\mu_1)(56\lambda^4 \\
&\quad +119\lambda^3\mu_1+162\lambda^2\mu_1^2+119\lambda\mu_1^3+56\mu_1^4)\mu_2^9+2(35\lambda^4+98\lambda^3\mu_1+131\lambda^2\mu_1^2 \\
&\quad +98\lambda\mu_1^3+35\mu_1^4)\mu_2^{10}+7(\lambda+\mu_1)(8\lambda^2+11\lambda\mu_1+8\mu_1^2)\mu_2^{11} \\
&\quad +2(14\lambda^2+25\lambda\mu_1+14\mu_1^2)\mu_2^{12}+8(\lambda+\mu_1)\mu_2^{13}+\mu_2^{14})
\end{aligned}
$$

$$
\begin{aligned}
f_2(\lambda,\mu_1,\mu_2,9) &= (\lambda+\mu_1)\mu_2^2(\lambda^7\mu_1^7(\lambda^2+\lambda\mu_1+\mu_1^2)+\lambda^6\mu_1^6(\lambda+\mu_1)^3\mu_2 \\
&\quad +\lambda^5\mu_1^5(\lambda+\mu_1)^4\mu_2^2+\lambda^4\mu_1^4(\lambda+\mu_1)^5\mu_2^3+\lambda^3\mu_1^3(\lambda+\mu_1)^6\mu_2^4+\lambda^2\mu_1^2(\lambda+\mu_1)^7\mu_2^5 \\
&\quad +\lambda\mu_1(\lambda+\mu_1)^8\mu_2^6+(\lambda+\mu_1)^9\mu_2^7+(9\lambda^8+44\lambda^7\mu_1+119\lambda^6\mu_1^2+209\lambda^5\mu_1^3 \\
&\quad +251\lambda^4\mu_1^4+209\lambda^3\mu_1^5+119\lambda^2\mu_1^6+44\lambda\mu_1^7+9\mu_1^8)\mu_2^8+(\lambda+\mu_1)(36\lambda^6+104\lambda^5\mu_1 \\
&\quad +197\lambda^4\mu_1^2+234\lambda^3\mu_1^3+197\lambda^2\mu_1^4+104\lambda\mu_1^5+36\mu_1^6)\mu_2^9+2(42\lambda^6 \\
&\quad +147\lambda^5\mu_1+273\lambda^4\mu_1^2+331\lambda^3\mu_1^3+273\lambda^2\mu_1^4+147\lambda\mu_1^5+42\mu_1^6)\mu_2^{10} \\
&\quad +14(\lambda+\mu_1)(9\lambda^4+20\lambda^3\mu_1+27\lambda^2\mu_1^2+20\lambda\mu_1^3+9\mu_1^4)\mu_2^{11} \\
&\quad +(126\lambda^4+364\lambda^3\mu_1+491\lambda^2\mu_1^2+364\lambda\mu_1^3+126\mu_1^4)\mu_2^{12}+12(\lambda+\mu_1)(7\lambda^2+10\lambda\mu_1 \\
&\quad +7\mu_1^2)\mu_2^{13}+(36\lambda^2+65\lambda\mu_1+36\mu_1^2)\mu_2^{14}+9(\lambda+\mu_1)\mu_2^{15}+\mu_2^{16})
\end{aligned}
$$

$$
\begin{aligned}
f_2(\lambda, \mu_1, \mu_2, 10) &= (\lambda + \mu_1)\mu_2^2(\lambda^8\mu_1^8(\lambda^2 + \lambda\mu_1 + \mu_1^2) + \lambda^7\mu_1^7(\lambda + \mu_1)^3\mu_2 \\
&+ \lambda^6\mu_1^6(\lambda + \mu_1)^4\mu_2^2 + \lambda^5\mu_1^5(\lambda + \mu_1)^5\mu_2^3 + \lambda^4\mu_1^4(\lambda + \mu_1)^6\mu_2^4 + \lambda^3\mu_1^3(\lambda + \mu_1)^7\mu_2^5 \\
&+ \lambda^2\mu_1^2(\lambda + \mu_1)^8\mu_2^6 + \lambda\mu_1(\lambda + \mu_1)^9\mu_2^7 + (\lambda + \mu_1)^{10}\mu_2^8 + (\lambda + \mu_1)(10\lambda^8 + 44\lambda^7\mu_1 \\
&+ 120\lambda^6\mu_1^2 + 209\lambda^5\mu_1^3 + 252\lambda^4\mu_1^4 + 209\lambda^3\mu_1^5 + 120\lambda^2\mu_1^6 + 44\lambda\mu_1^7 + 10\mu_1^8)\mu_2^9 \\
&+ (45\lambda^8 + 192\lambda^7\mu_1 + 462\lambda^6\mu_1^2 + 756\lambda^5\mu_1^3 + 887\lambda^4\mu_1^4 + 756\lambda^3\mu_1^5 + 462\lambda^2\mu_1^6 \\
&+ 192\lambda\mu_1^7 + 45\mu_1^8)\mu_2^{10} + 2(\lambda + \mu_1)(60\lambda^6 + 171\lambda^5\mu_1 + 312\lambda^4\mu_1^2 + 367\lambda^3\mu_1^3 \\
&+ 312\lambda^2\mu_1^4 + 171\lambda\mu_1^5 + 60\mu_1^6)\mu_2^{11} + 2(105\lambda^6 + 378\lambda^5\mu_1 + 707\lambda^4\mu_1^2 + 858\lambda^3\mu_1^3 \\
&+ 707\lambda^2\mu_1^4 + 378\lambda\mu_1^5 + 105\mu_1^6)\mu_2^{12} + 12(\lambda + \mu_1)(21\lambda^4 + 49\lambda^3\mu_1 + 66\lambda^2\mu_1^2 + 49\lambda\mu_1^3 \\
&+ 21\mu_1^4)\mu_2^{13} + 3(70\lambda^4 + 208\lambda^3\mu_1 + 283\lambda^2\mu_1^2 + 208\lambda\mu_1^3 + 70\mu_1^4)\mu_2^{14} + 3(\lambda + \mu_1)(40\lambda^2 \\
&+ 59\lambda\mu_1 + 40\mu_1^2)\mu_2^{15} + (45\lambda^2 + 82\lambda\mu_1 + 45\mu_1^2)\mu_2^{16} + 10(\lambda + \mu_1)\mu_2^{17} + \mu_2^{18})
\end{aligned}
$$

REFERENCES

[1] Ansell, P., Glazebrook, K., and Kirkbride, C., "Generalised join-the-shortest-queue policies for the dynamic routing of jobs to multi-class queues," *Journal of the Operational Research Society*, vol. 54, no. 4, pp. 379–389, 2003.

[2] Baccelli, F., Jean-Marie, A., and Mitrani, I., *Quantitative Methods in Parallel Systems*. Springer Science & Business Media, 2013.

[3] Balsamo, S., de Nitto Personé, V., and Onvural, R., *Analysis of Queueing Networks with Blocking*, vol. 31. Kluwer Academic Publishers, 2001.

[4] Bao, P., Pierce, J., Whittaker, S., and Zhai, S., "Smart phone use by non-mobile business users," in *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pp. 445–454, ACM, 2011.

[5] Bertsekas, D. and Gallager, R., *Data Networks*. Prentice-Hall, Inc, second ed., 1992.

[6] Bertsimas, D. and Tsitsiklis, J. N., *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA, 1997.

[7] Bonomi, F. and Kumar, A., "Adaptive optimal load balancing in a nonhomogeneous multiserver system with a central job scheduler," *IEEE Transactions on Computers*, vol. 39, no. 10, pp. 1232–1250, 1990.

[8] Boxma, O. J., Koole, G., and Liu, Z., *Queueing-theoretic solution methods for models of parallel and distributed systems*. Centrum voor Wiskunde en Informatica, Department of Operations Research, Statistics, and System Theory, 1994.

[9] Chang, K.-H. and Chen, W.-F., "Admission control policies for two-stage tandem queues with no waiting spaces," *Computers and Operations Research*, vol. 30, no. 4, pp. 589–601, 2003.

[10] Cheng, M. C., "New criteria for the simplex algorithm," *Mathematical Programming*, vol. 19, no. 1, pp. 230–236, 1980.

[11] Cheng, M. C., "Generalized theorems for permanent basic and nonbasic variables," *Mathematical Programming*, vol. 31, no. 2, pp. 229–234, 1985.

[12] Cheng, P. C., Rohatgi, P., Keser, C., Karger, P. A., Wagner, G. M., and Reninger, A. S., "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 222–230, IEEE, 2007.

[13] Clark, J. A., Tapiador, J. E., McDermid, J., Cheng, P.-C., Agrawal, D., Ivanic, N., and Slogget, D., "Risk based access control with uncertain and time-dependent sensitivity," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pp. 1–9, IEEE, 2010.

[14] Daskalaki, S. and Smith, J. M., "Combining routing and buffer allocation problems in series-parallel queueing networks," *Annals of Operations Research*, vol. 125, no. 1-4, pp. 47–68, 2004.

[15] de Nitto Persone, V. and Grassi, V., "Solution of finite QBD processes," *Journal of Applied Probability*, pp. 1003–1010, 1996.

[16] Denning, D. E. and MacDoran, P. F., "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud & Security*, vol. 1996, no. 2, pp. 12–16, 1996.

[17] Fan-Orzechowski, X. and Feinberg, E. A., "Optimality of randomized trunk reservation for a problem with a single constraint," *Advances in applied probability*, pp. 199–220, 2006.

[18] Fan-Orzechowski, X. and Feinberg, E. A., "Optimality of randomized trunk reservation for a problem with multiple constraints," *Probability in the Engineering and Informational Sciences*, vol. 21, no. 02, pp. 189–200, 2007.

[19] Feinberg, E. A. and Reiman, M. I., "Optimality of randomized trunk reservation," *Probability in the Engineering and Informational Sciences*, vol. 8, no. 04, pp. 463–489, 1994.

[20] Ghoneim, H. A. and Stidham, S., "Control of arrivals to two queues in series," *European Journal of Operational Research*, vol. 21, no. 3, pp. 399 – 409, 1985.

[21] Gordon, W. J. and Newell, G. F., "Closed queueing systems with exponential servers," *Operations Research*, vol. 15, no. 2, pp. 254–265, 1967.

[22] Gosavi, H. D. and Smith, J. M., "An algorithm for sub-optimal routeing in series-parallel queueing networks," *International Journal of Production Research*, vol. 35, no. 5, pp. 1413–1430, 1997.

[23] Grassmann, W. K. and Drekic, S., "An analytical solution for a tandem queue with blocking," *Queueing Systems*, vol. 36, no. 1-3, pp. 221–235, 2000.

[24] Guo, X. and Hernández-Lerma, O., *Continuous-time Markov Decision Processes.* Springer, 2009.

[25] Hordijk, A. and Kallenberg, L. C. M., "Linear programming and Markov decision chains," *Management Science*, vol. 25, no. 4, pp. 352–362, 1979.

[26] Hordijk, A. and Koole, G., "On the optimality of the generalized shortest queue policy," *Probability in the Engineering and Informational Sciences*, vol. 4, no. 04, pp. 477–487, 1990.

[27] HORDIJK, A. and KOOLE, G., "On the assignment of customers to parallel queues," *Probability in the Engineering and Informational Sciences*, vol. 6, no. 495-511, p. 3, 1992.

[28] HORDIJK, A. and KOOLE, G., "On the shortest queue policy for the tandem parallel queue," *Probability in the Engineering and Informational Sciences*, vol. 6, pp. 63–79, 1992.

[29] HUNT, P. and LAWS, C., "Asymptotically optimal loss network control," *Mathematics of Operations Research*, vol. 18, no. 4, pp. 880–900, 1993.

[30] JOHRI, P. K., "Optimality of the shortest line discipline with state-dependent service rates," *European Journal of Operational Research*, vol. 41, no. 2, pp. 157–161, 1989.

[31] KALLENBERG, L. C. M., "Finite state and action MDPs," in *Handbook of Markov Decision Processes*, pp. 21–87, Springer, 2002.

[32] KELLY, F. P., "Loss networks," *The Annals of Applied Probability*, pp. 319–378, 1991.

[33] KELLY, F. P., KEY, P. B., and ZACHARY, S., "Distributed admission control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2617–2628, 2000.

[34] KEY, P. B., "Optimal control and trunk reservation in loss networks," *Probability in the Engineering and Informational Sciences*, vol. 4, pp. 203–242, 4 1990.

[35] KIM, B. and KIM, J., "Optimal admission control for two station tandem queues with loss," *Operations Research Letters*, vol. 42, no. 4, pp. 257–262, 2014.

[36] KIM, C. and DUDIN, S., "Priority tandem queueing model with admission control," *Computers and Industrial Engineering*, vol. 61, no. 1, pp. 131 – 140, 2011.

[37] KOVED, L. and ZHANG, B., "Improving usability of complex authentication schemes via queue management and load shedding," in *Symposium on Usable Privacy and Security (SOUPS)*, Citeseer, 2014.

[38] KU, C.-Y. and JORDAN, S., "Access control to two multiserver loss queues in series," *IEEE Transaction on Automatic Control*, vol. 42, no. 7, pp. 1017–1023, 1997.

[39] KU, C.-Y. and JORDAN, S., "Access control of parallel multiserver loss queues," *Performance Evaluation*, vol. 50, no. 4, pp. 219 – 231, 2002.

[40] KU, C.-Y. and JORDAN, S., "Near optimal admission control for multiserver loss queues in series," *European Journal of Operational Research*, vol. 144, no. 1, pp. 166–178, 2003.

[41] LaBouef, K., "2016 update: What happens in one internet minute?." `http://http://www.excelacom.com/resources/blog/2016-update-what-happens-in-one-internet-minute/`, Blog post, 2016.

[42] Latouche, G. and Ramaswami, V., *Introduction to matrix analytic methods in stochastic modeling*, vol. 5. Siam, 1999.

[43] Leskelä, L. and Resing, J., "A tandem queueing network with feedback admission control," in *Network Control and Optimization*, pp. 129–137, Springer, 2007.

[44] Lewis, M. E., Ayhan, H., and Foley, R. D., "Bias optimality in a queue with admission control," *Probability in the Engineering and Informational Sciences*, vol. 13, no. 03, pp. 309–327, 1999.

[45] Lewis, M. E., Ayhan, H., and Foley, R. D., "Bias optimal admission control policies for a multiclass nonstationary queueing system," *Journal of applied probability*, pp. 20–37, 2002.

[46] Li, A. A. and Whitt, W., "Approximate blocking probabilities in loss models with independence and distribution assumptions relaxed," *Performance Evaluation*, vol. 80, pp. 82–101, 2014.

[47] Lin, K. Y. and Ross, S. M., "Admission control with incomplete information of a queueing system," *Oper. Res.*, vol. 51, no. 4, pp. 645–654, 2003.

[48] Lippman, S. A., "Applying a new device in the optimization of exponential queuing systems," *Operations Research*, vol. 23, no. 4, pp. 687–710, 1975.

[49] Litvak, N., van Rijsbergen, M., Boucherie, R. J., and van Houdenhoven, M., "Managing the overflow of intensive care patients," *European Journal of Operational Research*, vol. 185, no. 3, pp. 998–1010, 2008.

[50] Martin, J. B., "Large tandem queueing networks with blocking," *Queueing Systems*, vol. 41, no. 1, pp. 45–72, 2002.

[51] Miller, B. L., "A queueing reward system with several customer classes," *Management Science*, vol. 16, no. 3, pp. 234–245, 1969.

[52] Nagata, S. F., "Multitasking and interruptions during mobile web tasks," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 47, pp. 1341–1345, SAGE Publications, 2003.

[53] Ni, L. M. and Hwang, K., "Optimal load balancing in a multiple processor system with many job classes," *IEEE Transactions on Software Engineering*, no. 5, pp. 491–496, 1985.

[54] Nocedal, J. and Wright, S., *Numerical optimization.* Springer Science & Business Media, 2006.

[55] Paparrizos, K., Stephanides, G., and Samaras, N., "Improved criteria for identifying optimal basic and nonbasic variables in LP," *Journal of Computational Analysis and Applications*, vol. 3, no. 1, pp. 75–82, 2001.

[56] Perros, H. G., "Queueing networks with blocking: A bibliography," *ACM SIGMETRICS Performance Evaluation Review*, vol. 12, no. 2, pp. 8–12, 1984.

[57] Perros, H. G. and Altiok, T., "Approximate analysis of open networks of queues with blocking: Tandem configurations," *IEEE Transactions on Software Engineering*, no. 3, pp. 450–461, 1986.

[58] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[59] Reiman, M. I., "Optimal trunk reservation for a critically loaded link," in *Teletraffic and datatraffic in a period of change, ITC-13* (Jensen, A. and Iverson, V. B., eds.), pp. 247–252, 1991.

[60] Resnick, S. I., *Adventures in stochastic processes*. Springer Science & Business Media, 2013.

[61] Ross, K. W., "Randomized and past-dependent policies for markov decision processes with multiple constraints," *Operations Research*, vol. 37, no. 3, pp. 474–477, 1989.

[62] Sethuraman, J. and Squillante, M. S., "Optimal stochastic scheduling in multiclass parallel queues," *SIGMETRICS Performance Evaluation Review*, vol. 27, pp. 93–102, May 1999.

[63] Sheu, R.-S. and Ziedins, I., "Asymptotically optimal control of parallel tandem queues with loss," *Queueing Systems*, vol. 65, no. 3, pp. 211–227, 2010.

[64] Shi, Y., Yu, P.-L., and Zhang, D. Z., "Eliminating permanently dominated opportunities in multiple-criteria and multiple-constraint level linear programming," *Journal of Mathematical Analysis and Applications*, vol. 183, no. 3, pp. 685–705, 1994.

[65] Silva, D. F., Zhang, B., and Ayhan, H., "Optimal admission control for tandem loss systems with two stations," *Operations Research Letters*, vol. 41, no. 4, pp. 351–356, 2013.

[66] Silva, D. F., Zhang, B., and Ayhan, H., "Admission control strategies for tandem markovian loss systems." Submitted for publication to *Queueing Systems*, 2016.

[67] Silva, D. F., Zhang, B., and Ayhan, H., "Dynamic control of complex authentication systems." Submitted for publication to *IEEE Conference on Decision and Control*, 2016.

[68] Spicer, S. and Ziedins, I., "User-optimal state-dependent routeing in parallel tandem queues with loss," *Journal of Applied Probability*, vol. 43, pp. 274–281, 2006.

[69] Stidham Jr, S. and Weber, R., "A survey of Markov decision models for control of networks of queues," *Queueing Systems*, vol. 13, no. 1-3, pp. 291–314, 1993.

[70] Trafton, J. G., Altmann, E. M., and Brock, D. P., "Huh, what was i doing? how people use environmental cues after an interruption," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, pp. 468–472, SAGE Publications, 2005.

[71] Weber, R. R., "On the optimal assignment of customers to parallel servers," *Journal of Applied Probability*, pp. 406–413, 1978.

[72] Whitt, W., "Deciding which queue to join: Some counterexamples," *Operations research*, vol. 34, no. 1, pp. 55–62, 1986.

[73] Winston, W., "Optimality of the shortest line discipline," *Journal of Applied Probability*, pp. 181–189, 1977.

[74] Ye, Y., "Eliminating columns in the simplex method for linear programming," *Journal of Optimization Theory and Applications*, vol. 63, no. 1, pp. 69–77, 1989.

[75] Zhang, B. and Ayhan, H., "Optimal admission control for tandem queues with loss," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 163–167, 2013.

[76] Zhang, R. and Phillis, Y. A., "Fuzzy control of arrivals to tandem queues with two stations," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 361–367, 1999.

[77] Ziedins, I., "A paradox in a queueing network with state-dependent routing and loss," *Advances in Decision Sciences*, vol. 2007, pp. 1–10, 2007. (Article ID 68280).

VITA

Daniel F. Silva was born in Bogotá, Colombia. He holds B.Sc. degrees in Industrial Engineering and Mechanical Engineering from Universidad de Los Andes, in Colombia. In recognition for his academic achievements as an undergraduate, Universidad de Los Andes offered him a full scholarship for a Master's degree in Industrial Engineering, which he completed in 2008, graduating Cum Laude.

Before coming to Georgia Tech, Daniel worked as an Operations Research analyst at Kimberly Clark and Amazon.com for over four years, creating and implementing operations research solutions to a variety of supply chain, logistics and marketing problems. The solutions he worked on have been implemented by local operations in over 12 countries in three continents.

In 2011, Daniel joined the Ph.D. program for Operations Research at the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Tech. He carried out his dissertation research under the supervision of Dr. Hayriye Ayhan. Daniel's broader research interests involve developing and applying novel techniques to solve stochastic optimization problems arising in diverse applications, including supply chains, healthcare, and other service systems. He also develops software applications related to stochastic optimization. This includes working as a major contributor to the jMarkov Project, a Java framework for Markov chain modeling, which is currently a part of the COIN-OR initiative.

After graduation, Daniel plans to start working as an Assistant Professor in the Department of Industrial and Systems Engineering at Auburn University, starting in the fall of 2016.