Time Sensitive Operations

Mehmet Berat Aydemir

May 2021

Tepper School of Business
Carnegie Mellon University

Submitted In Partial Fulfillment Of The Requirements
For The Degree Of
Doctor Of Philosophy In
Operations Management
At Tepper School of Business, Carnegie Mellon University

Dissertation Committee:
Alan Andrew Scheller-Wolf (Co-Chair)
Alexandre Jacquillat (Co-Chair)
Mohammad Delasay
Mustafa Akan
Alan Montgomery

Abstract

Time is an essential component of many operations. Any good or service provided by its producers must be delivered on time. Advancements in information technology and urban development through years has only exacerbated the demand and importance of timely operations. For certain good and services being timely is a new feature that was not required before whereas for others being timely is at the core of their value proposition. This dissertation examines three problems that have arisen in response to these advancements in information technology and urban development.

In the first chapter, we study the dispatching problem faced by the dispatchers in Emergency Medical Services (EMS). Decision making in 911 dispatching of EMS plays a critical role in potentially saving patients' lives during time-sensitive emergencies. Getting resources to a patient as quickly as possible is essential to this task, and any delays can be life-threatening. Motivated by the challenges faced by the dispatchers in EMS, we develop a novel data-driven framework by combining ideas from Machine Learning and Optimization to guide dispatcher decisions. First, we propose a novel omniscient optimization model for ambulance dispatching that incorporates forward-looking decisions. Second, we employ Scenario Based Robust Optimization framework that utilizes ideas from Stochastic Programming and Robust Optimization where the uncertainty sets and associated scenarios are built using our novel *Closest Neighbours Clustering* method. This method first uses past data to select a set of similar calls to the current emergency call to build a set of future scenarios. Then it clusters the set of scenarios using the scenario metric we developed. These clusters are later fed into Scenario Based Robust Optimization model to produce a dispatch decision under future uncertainty. Experimental results show that our framework can improve the percentage of late responses as much as 25% compared to current dispatch methods.

In the second chapter, we study an on-demand platform's delay information disclosure policy when the platform serves two classes of users—consumers and providers—who seek matches to each other using the platform. We model the platform as maximizing the average rate at which these users are successfully matched by choosing one of three information regimes —- occupancy (disclosing the current system occupancy to both user classes), and two asymmetric information regimes (disclosing no information to one user class and occupancy information to the other). Arriving users are strategic and decide whether to join the system or not based on the delay information that the platform provides. In our base model, we consider users of each class as being either patient (will wait to

be matched) or impatient (will join only if they expect to be matched immediately). We find that depending on the parameter setting, any of the three information disclosure policies could emerge as optimal; however, the optimal policy has a complicated dependence on the parameters. We analytically establish sufficient conditions driving the platform's decision. We also examine two limiting settings: (i) patience profile discrepancy between the two user classes; and (ii) market size imbalance between the two user classes. For these two limiting settings, we characterize the platform's optimal information regime and show that all three disclosure policies may emerge in equilibrium. We then numerically examine the impact of the platform's choice of information regime on users' welfare and find that the platform's choice can also maximizes the welfare of both user classes, but only if this choice is to disclose occupancy information to both classes. We extend our base model to study how the platform's information regime choice changes when user patience levels are more heterogeneous.

In the third chapter, we study the problem of balancing efficiency and risk in multi-class screening systems. Security screening systems aim to identify malevolent people and illicit goods. But screening operations may also result in long wait times at checkpoints. Selecting appropriate screening procedures thus creates a trade-off between efficiency and risk. This is complicated by the heterogeneity of screening jobs (which pose various threat levels) and the strategic behaviors of human agents (who may renege prior to screening if perceived risk levels are too high). We apply a speed-quality trade-off perspective to security operations and extend the speed-quality trade-off literature to a multi-class setting with heterogeneous and strategic agents. From a practical standpoint, our work supports tactical decision-making for dynamically selecting screening procedures, and strategic decision-making for designing pre-screening profiling programs. We formulate this problem with continuous-time infinite-horizon Markov decision processes to optimize service rates in an $M/M/1$ queue with heterogeneous jobs, as a function of observed queue lengths and a threat level estimate for each job. We propose an extension to capture endogenous strategic behaviors of heterogeneous agents, given information asymmetries between agents and the screening operator. We show that the optimal policy exhibits a double threshold behavior: the shorter the queue length and/or the larger the risk, the stricter the screening. Leveraging job-level risk information can reduce expected costs by up to 6-7%, as compared to single-class decision-making schemes. Moreover, anticipating agents' strategic behaviors results in more intensive screening in an attempt to force malevolent agents to renege. Slower screening mitigates expected risks and may also, surprisingly, reduce expected queue lengths.

Acknowledgments

I would like to begin with expressing my sincere gratitude to my advisors Prof Alexandre Jacquillat and Prof Alan Scheller-Wolf. From the moment we have started working together, Alex has been an inspiring, uplifting and invigorating mentor. He has been a lighthouse in the deep waters of research and paramount to my growth as an academic. Alan has been a role model that I will always look up to for his wisdom, integrity and professionalism and I will always remember him with his friendly assuring smiles. Working with Alan and Alex has been the best part of my academic experience.

I thank my co-authors Prof. Mohammad Delasay and Prof. Siddharth Prakash Singh for their time, guidance and mentorship through my doctoral studies. Mohammad was my first mentor with whom I was very lucky to work with and who understood what I needed as a novice researcher. I will certainly miss our discussions and weekly meetings with Siddharth. I am deeply thankful to Dr Leonard Weiss for our collaboration in my recent work. His guidance and generosity in providing a great research environment has been most helpful in hard times.

I also thank other members of my doctoral committee, Prof Mustafa Akan and Prof Alan Montgomery. I deeply appreciate Mustafa's guidance and his sharing of research ideas with me. I appreciate the invaluable feedback and comments that Alan has given me on my thesis. I'm also grateful for the research environment provided by Prof. Sridhar Tayur, Prof. Sunder Kekre, Prof. Nicola Secomandi, Prof. Soo-Haeng Cho. I will cherish the insights of teaching which I have learned from Prof. Kekre. I cannot go without mentioning Prof. Isa Hafalir who helped me to a great extent in my first summer as a novice researcher.

I'm immensely delighted to be in the same cohort with my close friends and office mates Amin Hosseininasab, Franco Berbeglia, Bo Yang, Ilqar Ramazanli and Arash Haddadan. I'm particularly grateful to Amin, Franco and Ilqar with whom we shared long nights and days in discussing and solving research as well as real-life obstacles. My gratitude extends to larger student body in Tepper that includes: Leela Nageswaran, Neda Mirzaeian, Musa Celdir, Savannah Tang, Neha Neha, Gerdus Benade, Alex Kazachkov, Thiago Serra, Ozgun Elci, Sagnik Das, Zeynep Korkut, Violet Chen and Yuyan Wang.

My special thanks goes to Lawrence Rapp and Laila Lee who always provided the most friendly and ready assistance to us PhD students at Tepper.

I want to express my deepest gratitude to my family. I am who I am right now thanks to my

parents Meral and Birol and my siblings Elif and Burak. Any success of mine is also their success and nothing would be possible without their unconditioned love and support.

Lastly, there are no words to describe my gratitude to my wife Aysenur who stood by me day and night. She was the most understanding and patient in all the long nights, weekends and holidays that I had to work. This period of my life and the future would be incredibly hard without her love, affection and companionship.

Table of Contents

Chapter 0

Introduction

"Time waits for no one" – Folklore.

Time is an essential component of many operations. Any good or service provided by its producers must be delivered on time. Advancements in information technology and urban development through years has only exacerbated the demand and importance of timely operations. For certain good and services being timely is a new feature that was not required before whereas for others being timely is at the core of their value proposition. This dissertation examines three problems that have arisen in response to these advancements in information technology and urban development.

In the first chapter we tackle the problem faced by Emergency Medical Service dispatchers who assign ambulances to emergency calls for a timely response. Their decision play a critical role in the lives of patients. For example, in instances of cardiac arrests, for every minute without cardiopulmonary resuscitation (CPR - an emergency lifesaving procedure performed when the heart stops beating), the chance of patient's death increases by 10% (Larsen et al., 1993). In instances of motor vehicle crashes (MVC) - another common cause of need for EMS, longer response times were significantly associated with higher rates of mortality (Byrne et al., 2019). Accordingly, preventing late responses and improving response times in a EMS system are life-saving measures we aim to provider in this chapter.

In order to help decision making of dispatchers, we propose a data-driven approach combining principles from machine learning and optimization. We collaborated with the University of Pittsburgh Medical Center (UPMC) and Allegheny County EMS in building our solution approach, specifically, we utilize data on past emergency calls provided by the Allegheny County EMS. Given a current emergency call, our method leverages past emergency call data to form a set of representative scenarios using conditional clustering and feeds these scenarios into a scenario based robust optimization model to produce a dispatch decision for the current emergency call. We propose four contributions in this chapter: First, we develop a novel omnisicient optimization model for the ambulance dispatching problem that incorporates forward-looking decisions. Second, we model and solve the dispatch decision problem with future uncertainty by employing scenario-based robust optimization - a novel

approach combining principles of stochastic programming and robust optimization. Third, and in conjunction with the scenario based robust optimization framework, we introduce conditional clustering - a data-driven clustering approach for forming of the uncertainty sets. Last, through thorough experimentation, we show that our solution method to the EMS dispatching problem out-performs existing benchmarks and state-of-the art methods introduced in the literature.

In the second chapter, motivated by the rise of so-called "on-demand platforms", we study the information sharing problem faced by the managers of these platforms. On-demand platforms facilitate matches between two user classes: Consumers, who seek service through the platform, and providers, who supply the service to the consumers through the platform. An important operational tool for these platforms is to disclose delay information to their users to induce desirable joining/balking behaviors. By using information as an operational lever platforms maintain acceptable delays in the system and deliver timely services and products to their users. However, It is not immediately clear *how best* a platform can use the leverage of sharing delay information and this problem is further complicated by the fact that, there are various *types* of information to disclose.

We model the operations of an on-demand platform as a two-sided queue and we investigate the platform's optimal delay information disclosure policy when users on both sides of the market are delay-sensitive. This optimal policy seeks to maximize the average rate at which the platform successfully matches consumers with providers (*match rate*); we use the match rate as a proxy for the platform's revenue and profitability. We focus on three information regimes—the *occupancy information* where *both* consumers and providers are informed of the current occupancy level in the platform, and two *asymmetric* information regimes where *either* consumers or providers are provided with occupancy information, while the other side is provided with no information. We study how the choice of information regime affects the platform's profitability by analyzing and comparing the match rates resulting from employing each of these regimes. In our model, users obtain a reward for receiving service and incur a delay cost. Arriving users form an expectation of the delay cost they will incur based on the delay information provided to them by the platform. Based on this expectation, they decide whether to join the platform or balk in order to maximize their utility.

In summary we propose three contributions in chapter two: First, we model an on-demand platform as a two-sided queuing system that facilitates matches between two user classes. Users in each class are heterogeneous in their delay sensitivities. We contribute to the literature by studying the platform's information regime choice, that determines what information to disclose to users of *both* user classes. Second, we compute the effective match rates associated with each information regime. We characterize sufficient conditions for each regime to emerge in equilibrium and provide insight into the conditions under which the platform prefers to hide information from one user class. Third, as an extension, we demonstrate that for systems with higher levels of user heterogeneity (a larger number of distinct patience sensitivity levels), the platform prefers to provide occupancy information to at least one of its user classes for a wider range of parameters.

Motivated by the long delays experienced at security lines at airports, we study the problem

of balancing efficiency and risk in multi-class screening systems. In general, security screening systems at land borders, airports, seaports and other military and civil facilities aim to identify malevolent people and illicit goods to mitigate downstream risks. The global scale of these operations is enormous. For example, an estimated 1.1 million passengers cross US borders every day, resulting in the seizure of around 8,000 drugs, $300,000 worth of currency and $3.8 million worth of products in violation of Intellectual Property Rights laws on a typical day (US Customs and Border Protection, 2016). At the same time, screening operations often create long queues at checkpoints, which impose significant system-wide costs. Screening systems thus need to balance the objectives of risk management—i.e., ensuring reliable operations to identify threats—and operating efficiency—i.e., maintaining appropriate screening speeds to avoid long wait times.

In this chapter, we formulate decision-making models to select screening procedures in multi-class settings. It takes the viewpoint of a screening operator who must balance efficiency and risk by selecting one procedure for each job (person or package) from a menu of available ones. We first consider instances with non-strategic agents, to determine how the selection of screening procedures can leverage real-time information on queue lengths and each job's perceived threat level. We then extend our model to determine how anticipating agents' strategic decisions to go through screening or renege may impact the optimal selection of screening procedures. In both instances, we identify the value of pre-screening information with respect to expected queue lengths and risk costs. More specifically, we propose the following contributions in chapter three: First, we introduce a novel dynamic decision-making model to balance efficiency and risk in multi-class queuing systems and show that the optimal policy exhibits a double threshold behavior: All else equal, the shorter the queue length and/or the higher the risk level, the stricter the optimal screening. Second, we find that system performance increases by up to 6-7% if the pre-screening risk information is utilized in the decision making. Third, we extend the base model to account for possible strategic behavior of screened agents. Fourth, we assess the impact of strategic behavior and pre-screening information on optimal screening policies and system performance. We find that agents' strategic behaviors results in stricter screening to try to deliberately force malevolent agents to renege. Moreover, we find that better pre-screening information reduces total expected costs by mitigating information asymmetries; however it does not necessarily reduce expected risks.

Chapter 1

Data-Driven Dispatch Decisions for Emergency Medical Services

1.1 Introduction

Decisions of dispatchers in Emergency Medical Services (EMS) play a critical role in saving lives of patients. For example, in instances of cardiac arrests, for every minute without cardiopulmonary resuscitation (CPR - an emergency lifesaving procedure performed when the heart stops beating), the chance of the patient's death increases by 10% (Larsen et al., 1993). It only takes 6-10 minutes without care for irreversible brain cell damage and death (Weinberger et al., 1940). In instances of motor vehicle crashes (MVC) or other traumatic injuries, which are other common triggers for EMS response, longer response times were significantly associated with higher rates of mortality (Byrne et al., 2019; Sampalis et al., 1999). Accordingly, preventing late responses and improving response times in an EMS system are life-saving measures that many researchers have worked on in various disciplines (Aringhieri et al., 2017).

An important component of an EMS system are the dispatchers who assign ambulances to incoming emergency calls. EMS dispatchers face a complex problem as the decisions they make impact the response time for the current call as well as the future incoming emergency calls. For instance, if the only ambulance that is covering a neighbourhood is dispatched to an emergency call that can be reached by another ambulance in time, this leaves the neighbourhood in a dire situation for (possibly) incoming new emergency calls. This highlights the significant challenge in dispatchers' decision making under future uncertainty of when and where next emergency calls will arrive. This challenge is made more difficult by the fact that prediction of spatial and temporal components of events such as emergency calls are inherently difficult. While general trends of when and where emergency calls arrive can be driven using past data, utilization of these trends on the tactical level is not straightforward as these trends typically lack precision and do not incorporate the current state of the system.

Although there is no national standard for pure EMS response, a common measure of performance for EMS systems is often guided by National Fire Protection Association's (NFPA) Standard 1710, recommending the response time of a first responder to be less than 4 minutes from the time of dispatch 90% of the time (Association et al., 2010). In order to achieve this goal, we propose a data-driven approach combining principles from machine learning and optimization to tackle the problem faced by the EMS dispatchers on the tactical level. We collaborated with the University of Pittsburgh Department of Emergency Medicine and the City of Pittsburgh Bureau of EMS in designing our solution approach, specifically, we utilized anonymized retrospective emergency calls in the locale between the date of 2017-2018. Given a current emergency call, our method leverages past emergency call data to form a set of representative scenarios using Closest Neighbours Clustering and feeds these scenarios into a scenario based robust optimization model to produce a dispatch decision for the current emergency call.

The first component of our method is a novel omniscient deterministic binary linear program that considers forward looking dispatch decisions. In general omniscient models aim to solve the dispatching problem under perfect future information. However, current state-of-the art omniscient models do not consider forward-looking dispatch decisions that allow for a busy ambulance to be queued in response to a new emergency call. We incorporate this feature in a new omniscient model that we refer to as a Deterministic-Binary Linear Program (D-BLP) which is used as the basis of our solution method for the dispatching problem under future uncertainty (the locations and timings of emergency calls yet to arrive). To model future uncertainties, we utilize the past emergency call data to form representative future scenarios. We refer to this first step of our solution method as Closest Neighbours Clustering which consists of three stages: (i) Identification of Close Calls, (ii) Formation of Scenario Sets, (iii) Clustering of Scenarios. Once we have a set of representative future scenarios we then solve a scenario-based robust optimization model to obtain the dispatch decision for the current emergency call.

Scenario-based robust optimization is a new modelling approach developed by Wang and Jacquillat (2020). It combines ideas from stochastic programming and robust optimization by considering probabilistic scenarios that are adversely chosen from corresponding uncertainty sets. This approach neatly fits into our framework, as each representative future scenario is chosen from a cluster of future scenarios that was built in the Closest Neighbours Clustering step. To solve the scenario-based robust optimization model we employ the cutting plane algorithm proposed in Wang and Jacquillat (2020), we show that employing this algorithm improves the solution time of our formulation by around 60%, yielding practical solution times.

We compare our approach to a series of benchmarks from the literature such as the closest dispatch policy (CDP) and dynamic-maximum-expected-coverage-location heuristic for dispatching (DMEXCLP-D). Our method's performance in terms of number of late calls achieves approximately a 25% improvement over DMEXCLP-D and a 60% improvement over closest dispatch policies. We also propose several variants of our approach as benchmarks to investigate the benefits of Closest

Neighbours Clustering, multiple and adverse scenario considerations.

This chapter makes the following contributions. First, we develop a novel omnisicient optimization model for the ambulance dispatching problem that incorporates forward-looking decisions. Second, we model and solve the dispatch decision problem with future uncertainty by employing scenario-based robust optimization - a novel approach combining principles of stochastic programming and robust optimization. Third, and in conjunction with the scenario based robust optimization framework, we introduce Closest Neighbours Clustering - a data-driven clustering approach for forming of the uncertainty sets. Last, through thorough experimentation, we show that our solution method to the EMS dispatching problem out-performs existing benchmarks and state-of-the art methods introduced in the literature.

1.2 Literature Review

Our work is related to two streams of literature (i) emergency medical services (ii) data-driven optimization.

Emergency Medical Services: A variety of strategic and tactical problems faced in EMS has received significant attention from researchers from the field of management science. Readers can refer to recent surveys by Aringhieri et al. (2017) and Bélanger et al. (2019) for a general discussion of the literature. On the strategic level a well-known topic addressed by researchers is the ambulance station location problem which is a special case of the facility location problem (see Li et al. (2011), Başar et al. (2012), Ahmadi-Javid et al. (2017) and Güneş et al. (2019) for a survey of studies). On a tactical level, the problem of where to locate/deploy ambulances that finish serving patients has been addressed as a problem of relocation/redeployment by following series of papers: Brotcorne et al. (2003), Gendreau et al. (2006), Andersson and Värbrand (2007), Maxwell et al. (2010), Alanis et al. (2013), Saydam et al. (2013), Sudtachat et al. (2016), Van Barneveld (2016), Van Barneveld et al. (2017). Few papers (Schmid (2012) and Nasrollahzadeh et al. (2018)) also combine the decision of dispatching into their model of relocation, and compare their results with a series of benchmarks. Using approximate dynamic programming they show that the implementation of reallocation and redeployment policies leads to improvements in the system while the inclusion of dispatching decisions into their models provide small improvements when compared to the closest dispatch policy. In contrast to these papers, we only focus on the problem of dispatching, and assume that finishing ambulances return back to their bases, which replicates the current practice. Moreover, when compared to the closest dispatch policy our method provides more significant improvements in the performance of the system.

Among papers that focus only on dispatching, Carter et al. (1972), Bandara et al. (2012) McLay and Mayorga (2013a,b), Bandara et al. (2014) Jagtenberg et al. (2017a) and Sudtachat et al. (2014) propose heuristic, simulation and Markov decision process based dispatching models with various features such as objectives of patient survivability, coverage level, severity of calls, and different

types of medical units to show the sub-optimality of closest dispatch policies. A close work to ours is Jagtenberg et al. (2017b), in which authors propose an offline optimization model in which future emergency calls are known in advance; they compare the performances of this offline optimization, the closest-dispatch policy and the DMEXCLP-D heuristic proposed in Jagtenberg et al. (2017a). Based on experiments using data from a large ambulance provider in Netherlands, they show that the closest dispatch policy is 2.7 times worse than the offline optimal policy in terms of the fraction of late calls, whereas this number is dropped down to 1.9 times with the DMEXCLP-D heuristic. We compare our results with that of DMEXCLP-D heuristic and show that we can improve it by another 15%.

Data-Driven Optimization: With advancements in information technology, the collection, availability and utilization of data in business practices have gained significant momentum. Accordingly, data-driven optimization has drawn attention from management science researchers. Readers can refer to Bastani et al. (2020) for a recent and broad review of the topic. The majority of related work first makes predictions using supervised learning and then feeds these predictions into various optimization models ("predict then optimize"). For instance, in the context of supply chain management a series of papers by Gallien et al. (2015), Ferreira et al. (2016), Alley et al. (2019), Aouad et al. (2015), and Glaeser et al. (2019) first employ forecasting and then use the outcome of forecasting in optimization of decisions such as shipment, pricing, assortment and facility location, respectively. In the context of urban planning, Bertsimas et al. (2019) and Liu et al. (2020) use supervised learning to predict travel times for school bus routing and last mile delivery.

Beginning with Bertsimas and Kallus (2020), researchers have started to explore incorporating the prediction step into optimization models. More specifically, Bertsimas and Kallus (2020) employ a weighted objective for a stochastic optimization problem and derive the associated weights using supervised learning methods. In follow up work, Elmachtoub and Grigas (2021) generalize this framework by introducing "Smart Predict then Optimize" (SPO), where the loss function of the learning step considers the decision error induced by the prediction at the optimization step. Balghiti et al. (2019) provides generalization bounds. Mandi et al. (2020) and Elmachtoub et al. (2020) extend the SPO framework to solve large-scale combinatorial optimization problems and tree based prediction models respectively.

A closely related work to ours is by Wang and Jacquillat (2020), in which the authors propose the scenario based robust optimization framework for solving problems under categorical uncertainty. They also develop a cutting plane algorithm for solution of the scenario based robust optimization problem. We use this approach and algorithm for our work.

While the utilization of supervised learning is prevalent -as seen in the above literature- clustering is another methodology used in Machine Learning. To our knowledge, only notable work that employs clustering for estimation of features in optimization is by Bernstein et al. (2019), in which the authors propose a dynamic clustering policy for estimating customer preferences for personalized assortment optimization.

In contrast to the above mentioned literature, we are first to employ clustering as a tool of preparing the data to feed into the scenario based robust optimization step, and we are first to employ such a technique in the context of emergency medical services.

1.3 Notation, Data and Normalizations

Throughout the chapter, we denote the emergency calls with letter c. Each emergency call has three features, longitude, latitude and time of arrival which we denote with x, y and t, respectively, i.e., $c = (x, y, t)$.

The data set contains information of emergency calls from January 1st 2018, through February 28, 2019. The information about the calls includes the location in terms of longitude & latitude and date & time of arrival. The data-set includes a total of 39,957 emergency calls and it is denoted with $\mathcal{P} := \{c_1, c_2, \ldots, c_{39,957}\}$. We use the initial six month of this data as the training set denoted with $\mathcal{P}^{Tr}$ and the remaining seven months as the test set denoted with $\mathcal{P}^{Ts}$.

We apply a normalization to the data in order to better construct scenarios and relate the magnitude of effects between the location and time variables of the emergency calls. Specifically, we take the projection of latitude, longitude and time of arrival variables into a simpler three dimensional space. The projection of longitude and latitude are rather straightforward: We simply project each x and y into the $(0, 1)$ interval by subtracting x_{min} and y_{min} and then multiplying with ($1/x_{max}$ and $1/y_{max}$) where:

$$
\begin{aligned}
x_{min} &= \min_{i \in \{1,\ldots,39,957\}} x_i \\
y_{min} &= \min_{i \in \{1,\ldots,39,957\}} y_i \\
x_{max} &= \max_{i \& j \in \{1,\ldots,39,957\}} (|x_i - x_j|) \\
y_{max} &= \max_{i \& j \in \{1,\ldots,39,957\}} (|y_i - y_j|)
\end{aligned}
$$

We normalize time in two steps. First, we translate the time of arrival of each call to the time of arrival measured from the beginning of the day. We start each day at 4 am as, this is the time frame that has the least amount of activity. For instance, if an emergency call arrived at 6:05 am, its time of arrival would be translated into 125. Then at the second step, we project this value into a $(0, 1)$ interval by dividing it by 1440 (=24*60).

We denote an ambulance with the letter v; each ambulance is associated with a base. We denote the set of ambulance bases with $\mathbb{B} = \{1, 2, ..., B\}$ where B is the number of ambulance bases. For ease of exposition we assume that each ambulance base has exactly single ambulance, i.e., $v \in \mathbb{B}$. Whenever an ambulance responds to a call, it is dispatched from its base location. Accordingly, the travel time associated with a dispatch decision is calculated using the locations of the ambulance

base and the emergency calls. We denote this travel time with $\psi(c, v)$.

1.4 Closest Neighbours Clustering:

The first step of our approach is the formation of the future projections that will be fed into the optimization model. This is the data-driven step that utilizes the training set. We refer to this step as Closest Neighbours Clustering and apply it in three stages:

1. Identification of Close Calls: Given an emergency call c, find the N closest calls in $\mathcal{P}^{Tr}$. We denote this set as $\mathcal{P}_N(c)$.

2. Formation of Scenario Sets: Using the set $\mathcal{P}_N(c)$ we construct a set of scenarios. We denote the set of scenarios with $\mathcal{S}(c)$.

3. Clustering of Scenarios: We cluster the scenario set $\mathcal{S}(c)$ into K subsets $\mathcal{S}^k(c)$ where $k \in \mathcal{K} = \{1, 2, \ldots, K\}$ and $\mathcal{S}^k(c) \cap \mathcal{S}^{k'}(c) = \emptyset$, $\forall k, k' \in \mathcal{K}$ and $\cup_{k \in \mathcal{K}} \mathcal{S}^k(c) = \mathcal{S}(c)$.

The rationale behind this procedure is to form representative future scenarios from the training set. In the first stage, we identify similar emergency calls for a given current call; these calls will form the starting point of each scenario. Once the scenario set is obtained, we apply clustering to obtain a partition of the larger scenario set. Each partition is a subset of scenarios that look similar to each other. As part of the optimization, one member from each cluster is adversely chosen as a representative future scenario. By doing so, the optimization step avoids consideration of similar scenarios in the larger scenario set while protecting against worst case situations. Below we explain each stage of Closest Neighbours Clustering in more detail.

1.4.1 Identification of Close Calls

In this stage, we identify the N emergency calls that look most similar to the current emergency call in request of an ambulance. In order to assess similarity, we define a metric that measures how similar different emergency calls are to each other. This metric takes into account all the components of a call, i.e., it incorporates both the spatial and temporal similarities between the calls. More formally, we refer to this metric as the between-calls-metric and denote it with $d^c(i, j)$, for any two calls c_i and c_j. We define it as:

$$d^c(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha(t_i - t_j)^2},$$

where α is weight parameter for the temporal component. We set $\alpha = 1$ for the remainder of the chapter.

Using the between-calls-metric we identify the N closest calls to the current emergency call and construct the set $\mathcal{P}_N(c) = \{c_1, c_2, \ldots, c_N\}$. Note that N is a parameter we choose; thus plays an

important role on the performance of the optimization step. If this parameter is chosen to be too small, then possibly informative future scenarios may be omitted from the model (under-fitting). If it is chosen too large, possibly irrelevant future scenarios may be included in the optimization and blur the decision making (over-fitting). This is similar to the well-known bias-variance trade-off in the machine learning literature.

1.4.2 Formation of Scenario Sets

A scenario is a set of temporally ordered emergency calls. We form each scenario by taking a call $c_i \in \mathcal{P}_N(c)$ and its successors over the next T minutes from the training set. Accordingly, we denote a scenario that begins with a call $c_i \in \mathcal{P}_N(c)$ as $s_i(c) := \{c_i^0, c_i^1, c_i^2, \ldots, c_i^{M_i}\}$, where $c_i^1, c_i^2, \ldots, c_i^{M_i}$ are the successors of the c_i in over the next T time units and in an ordered manner. The call c_i^0 corresponds to c_i. Note that different scenarios can have different number of calls, i.e., M_i can take different values for different i. Accordingly, by collecting these scenario starting with each call $c_i \in \mathcal{P}_N(c)$, we form the set of scenarios, i.e., $\mathcal{S}(c) := \{s_1(c), s_2(c), \ldots, s_N(c)\}$.

Notice that T is another parameter of the model that determines how forward looking scenarios are. Similar to the choice of N, choice of too small a T can lead to under-fitting, while choice of too large a T can lead to over-fitting.

1.4.3 Clustering of Scenarios

Once the scenario set is defined we then cluster the scenario set into subsets of similar scenarios. In order to do so, we introduce a new metric that measures the similarity between different scenarios. We refer to this metric as the between-scenarios-metric and denote it with $D^s(i,j)$ for distance between two scenarios $s_i(c), s_j(c) \in \mathcal{S}(c)$. Before formally defining the between-scenario-metric we order the set of scenarios $\mathcal{S}(c)$ for ease of exposition: We order the indices of scenarios in $\mathcal{S}(c)$ such that the number of calls in scenarios decrease as the scenario index increases. In other words, the scenario with the highest number of calls is indexed with 1 and the scenario with the lowest number of calls is indexed with N. In case of a tie, any arbitrary rule can be applied and does not effect the outcome of the clustering nor the distance metric. With the given ordered scenario set we now formally define the between-scenario-metric for scenarios $i < j$:

$$D^s(i,j) := \max_{c_m \in s_i(c)} \big(\min_{c_n \in s_j(c)} d^s(c_m, c_n) \big)$$

The between-scenario-metric measures the distance between two scenarios by first matching the closest calls from the smaller scenario for each call in the larger scenario and then taking the maximum of the shortest distances over the matched calls in the larger scenario. In other words, the distance between two scenarios is the largest distance among the distances between the closest calls of the two scenarios. This metric ensures that if two scenarios contain similar calls, the scenarios are

close to each other due to selection of closest calls from the scenarios. On the other hand if the two scenarios are not similar due to a dissimilar emergency call in one of the scenarios, this is expressed in the distance between scenarios by the farthest distance considered among the matched calls. In Proposition 1.1 we show that our between-scenario-metric $D^s(i,j)$ is a proper metric. The proof is presented in Appendix A.1.1.

Proposition 1.1. *$D^s(i,j)$ is a proper metric.*

Using the between-scenarios-metric, we create a distance matrix for a given set of scenarios, which we give to an off-the-shelf algorithm for partitioning of the scenario set. We employ the Partitioning Around Medoids (PAM) Algorithm which is a modified version of the well-known K-Means algorithm. While the K-means algorithm finds the center of a cluster as the mean of the members of the clusters, the PAM algorithm picks a member of the cluster as the cluster center. It yields a partition of the scenario set $\mathcal{S}(c)$ into K subsets $\mathcal{S}^k(c) = \{s_1^k(c), s_2^k(c), \ldots, s_{N(k)}^k(c)\}$ such that $k \in \mathcal{K} = \{1, 2, \ldots, K\}$ and $\mathcal{S}^k(c) \cap \mathcal{S}^{k'}(c) = \emptyset$, $\forall k, k' \in \mathcal{K}$ and $\cup_{k\in\mathcal{K}} \mathcal{S}^k(c) = \mathcal{S}(c)$. $N(k)$ represents the number of scenarios in cluster k. K is the number of clusters and a parameter of the model. An advantage of using the PAM algorithm is that it allows us to pick the center of each cluster as a representative scenario for specials cases of the optimization step.

While having a greater number of clusters improves the quality of "clustering" i.e., it decreases total within cluster distances, it does not necessarily improve the performance the optimization step. Similar to the parameters N and T, one might over-fit or under-fit the model if the choice of K is not correct.

At the final stage of clustering we form a probability distribution over the clusters as we will use this probability distribution in the optimization step. We denote the probability associated with each cluster with $p^k \in (0,1)$ $\forall k \in \mathcal{K}$. We set the probability of each cluster to be the number of scenarios in the cluster normalized by the total number of scenarios in the scenario set N, i.e., $p^k = \frac{|\mathcal{S}^k(c)|}{|\mathcal{S}(c)|} = \frac{N(k)}{N}$.

1.5 Optimization Model for Dispatch Decision

We develop a scenario based robust optimization model to determine which ambulance to dispatch for a given emergency call. In making this decision, we utilize possible future scenarios that we derive in Section 1.4.

Before formally defining the model, we introduce following assumptions which are also used in the literature (see Jagtenberg et al. (2017b)):

Assumption 1: The occurrence of emergency calls are independent of previous incidents. This is a realistic assumption that can also be verified in the data.

Assumption 2: The travel time between any two points is deterministic and known in advance. This is a realistic assumption for many cities in developed countries as ambulances have priority in

traffic and hence are not affected by the traffic congestion.

Assumption 3: The busy time of an ambulance excluding the travel time is known, deterministic and the same for all emergency calls. While this assumption seems limiting, it does not impact the comparison of different algorithms as we use this assumption in all of the benchmarks.

We begin by introducing a novel deterministic binary linear program (D-BLP) that determines the dispatch decisions for all the calls in a single scenario. This model produces an omniscient policy of dispatches; this dispatch policy serves as the upper bound on the performance of any dispatch policy for any given scenario. Then, using the deterministic binary linear program as a basis, we introduce a scenario based robust optimization model that incorporates the future uncertainty into the decision making.

In all the methods we consider, we aim to minimize the fraction of late calls. However, our formulation allows consideration of other objectives, such as minimization of average response time with minimal modifications.

1.5.1 Deterministic Binary Linear Program:

There are two inputs to the D-BLP that must be computed given a real-life scenario $\mathcal{S} = \{c_1, c_2, \ldots, c_S\}$. The first input is the penalty term for the objective function. Let $l_{cc^2v} \in \{0, 1\}$ denote the penalty associated with assigning ambulance v to call c after state c^2. If the ambulance arrives within the specified threshold time, l_{cc^2v} is set to 0 and 1 otherwise. The set of states for a call c_i that an ambulance can respond from consists of $c^2 \in C^2(c_i) := \{A, c_1, c_2, \ldots, c_{i-1}\}$, where $\{c_1, c_2, \ldots, c_{i-1}\}$ corresponds to emergency calls before call c_i and A corresponds to the availability state. To put it in another way, D-BLP model considers the dispatch of a busy ambulance to a new emergency call c_i while it is responding to a prior emergency call c^2. This is in contrast to the offline BLP model of Jagtenberg et al. (2017b), where authors assume that only available ambulances can be dispatched for an emergency call. We note that, for a given scenario of emergency calls, we trivially pre-compute l_{cc^2v} by calculating the response time of an ambulance v for an emergency call c using the location information. In doing so, we also account for the delay that is experienced when ambulance v responds to call c after responding to call c^2 if the response time frames of calls c and c^2 overlap. Response time frame is the time span that it takes for an ambulance to reach the location of the call, take care of the patient and become available at its base again. If response time frames of two calls overlap for an ambulance, then the ambulance has to finish serving the prior call before responding to the latter emergency call. Other penalty terms such as response time for a call c can be easily utilized by setting l_{cc^2v} appropriately.

The second input is the indicator term for the logical flow of the ambulances. Let $a_{cc^2v} \in \{0, 1\}$ denote the indicator term that is set to 1 if the response time-frames of calls c and c^2 for ambulance v overlap and 0 otherwise. In our experiments, we start a day from a time that all ambulances are available and hence we assume the indicator term for the availability state, a_{cAv} always takes value 1

for the D-BLP model. We later relax this assumption for our scenario based robust optimization model. Note that similar to the penalty term l_{cc^2v} we trivially pre-compute indicator term a_{cc^2v} using the arrival times of calls c and c^2 along with response time of ambulance v to these calls.

We note that in the calculation of l_{cc^2v} and a_{cc^2v} we only consider single overlaps, disregard the possibility of multiple overlaps. For example, if ambulance 1 is responding to calls X, Y and Z in this order with overlapping time frames of calls X-Y and Y-Z, the delay associated with responding to call Y after call X is disregarded in the calculation of terms l_{ZY1} and a_{ZY1}. As a remedy for this, we inflate the response times by a certain amount for overlapping response time frames when calculating the penalty term l_{cc^2v}. While we try for different values for the inflation of the response time under overlaps, we find 9 minutes to work well for time threshold of 9 minutes in the experiments we run. Altogether, these calculations lead to an approximation of dispatch decisions for a real-life scenario, but provide us with dispatch policies that serve as upper bounds for any online dispatch method as will be seen in the experimental results.

The decision variables representing the dispatch decisions are denoted with $x_{cc^2v} \in \{0,1\}$. They take value 1 if ambulance v is dispatched to emergency call c after state c^2 and 0 otherwise.

$$\min_{x_{cc^2v}} \sum_{c\in\mathcal{S}} \sum_{c^2\in C^2(c)} \sum_{v\in B} l_{cc^2v} x_{cc^2v} \tag{1.1}$$

$$s.t. \sum_{c^2\in C^2(c)} \sum_{v\in B} x_{cc^2v} = 1, \qquad c \in \mathcal{S} \tag{1.2}$$

$$\sum_{c^2\in C^2(c)} x_{c'c^2v} + a_{cc'v} + \sum_{c^2\in C^2(c) c^2\neq c'} x_{cc^2v} \leq 2, \qquad v \in B,\ c, c^{'} \in \mathcal{S}\ c > c^{'} \tag{1.3}$$

$$x_{cc^2v} \leq a_{cc^2v} \qquad v \in B,\ \ c \in \mathcal{S},\ c^2 \in C^2(c),\ c \neq c^2 \tag{1.4}$$

$$x_{cc^2v} \in \{0,1\} \tag{1.5}$$

Constraint (1.2) ensures that every emergency call $c \in \mathcal{S}$ is dispatched an ambulance. Constraint (1.3) ensures that if ambulance v were to respond to both emergency calls c and $c^{'}$ ($c^{'}$ being the prior one) and if the response time frames of two calls were to overlap, then the decision variable $x_{cc'v}$ is set to 1. Similarly, constraint (1.4) ensures that decision variable x_{cc^2v} is set to 0 if there are no overlaps between a call c and its prior call c^2. Together, constraints (1.3)-(1.4) capture the a given decision's impact on the subsequent calls and ensure that decision variables are set to correct values for the movement of ambulances.

As mentioned earlier, in comparison with the offline BLP model of Jagtenberg et al. (2017b), our D-BLP model considers forward-looking dispatch decisions which allows the D-BLP model to produce more flexible dispatch policies. In Proposition 1.2, we formalize this claim. Let us denote the feasible decisions space of offline BLP and D-BLP with X^{BLP} and X^{D-BLP}, respectively.

Proposition 1.2. *Every feasible decision in offline BLP is a feasible decision in D-BLP:*

$$X^{BLP} \subset X^{D-BLP}$$

In light of Proposition 1.2 an important advantage of D-BLP is that it does not require an assumption of availability of at least one ambulance whenever an incident occurs (which is assumed by Jagtenberg et al. (2017b)). In situations that violate this assumption the offline BLP of Jagtenberg et al. (2017b) becomes infeasible, whereas our model produces a solution.

The D-BLP model finds a dispatch policy for a given scenario. In other words, we find the dispatch policy for a set of calls in a specific time such as a day. In doing so, D-BLP assumes perfect knowledge of the future and hence provides an upper bound on the performance of dispatch methods we consider. Due to the consideration of only two-way overlaps, we cannot guarantee if the dispatch policy provided by D-BLP is an *optimal* policy for a given scenario, but as presented in Section 1.7, its results serve as an upper bound. We next present the scenario based robust optimization formulation which we build on D-BLP.

1.5.2 Scenario Based Robust Optimization Model

In this section we introduce our scenario based robust optimization model (SBROM) that incorporates future uncertainty when making a dispatch decision for a given emergency call c. SBROM builds upon D-BLP and aims to solve the dispatch decision by considering multiple scenarios chosen from corresponding uncertainty sets.

Note that SBROM outputs a dispatch decision for each emergency call c_i in a real-life scenario $s = \{c_1, c_2, \ldots, c_S\}$. Accordingly, it must be run S times to form the dispatch policy for real-life scenario s. This is done in a rolling horizon manner, i.e., dispatch decision is made for calls $c_1, \ldots, c_S$ in an ordered fashion. Accordingly, when solving for a call c_i an ambulance v might be busy. In order to account for this, we add a busy state B to the set of states, i.e., $C^2(c_i) := \{c_1, c_2, \ldots, c_{i-1}, A, B\}$.

The general idea in SBROM is that a cluster $\mathcal{S}^k(c_i) = \{s_1^k(c_i), s_2^k(c_i), \ldots, s_{N(k)}^k(c_i)\}$ for $k \in \mathcal{K}$ (formed by Closest Neighbours Clustering for the emergency call $c_i \in s$) forms a future forecast for the call c_i. For the purposes of robust optimization, each cluster forms an uncertainty set. SBROM aims to make a dispatch decision for call c_i given K possible future projections where each future projection is selected from the cluster $\mathcal{S}^k(c_i)$ as the worst possible projection. Naturally, each projection does not have equal weight on the decision as some projections can be more likely to occur (as some clusters are larger than others). Accordingly, the weight of each projection is the corresponding probability p^k of the cluster to which it belongs. Recall that p^k was found at the end of the Closest Neighbours Clustering step.

Our formulation is

$$\min_{x^{kn}_{cc^2v}} \sum_{k\in\mathcal{K}} p^k \max_{n\in\{1,\ldots,N(k)\}} \sum_{c\in C}\sum_{c^2\in C^2}\sum_{v\in B} l^{kn}_{cc^2v}x^{kn}_{cc^2v} \tag{1.6}$$

$$s.t. \sum_{c^2\in C^2}\sum_{v\in B} x^{kn}_{cc^2v} = 1, \qquad c\in C,\ k\in\mathcal{K} \tag{1.7}$$

$$\sum_{c^2\in C^2} x^{kn}_{c^{'}c^2v} + a^{kn}_{cc^{'}v} + \sum_{c^2\in C^2 c^2\neq c^{'}} x^{kn}_{cc^2v} \leq 2, \qquad v\in B,\ c,c^{'}\in C\ c>c^{'},\ k\in\mathcal{K} \tag{1.8}$$

$$x^{kn}_{cc^2v} \leq a^{kn}_{cc^2v} \qquad v\in B,\ \ c\in C,\ c^2\in C^2 c\neq c^2,\ k\in\mathcal{K} \tag{1.9}$$

$$\sum_{c^2\in C^2} x^{kn}_{1c^2v} = \sum_{c^2\in C^2} x^{n^{'}k^{'}}_{1c^2v} \qquad v\in B,\ \ c\in C,\ k,k^{'}\in\mathcal{K} \tag{1.10}$$

$$x^{kn}_{cc^2v}\in\{0,1\} \tag{1.11}$$

Similar to D-BLP the inputs $l^{kn}_{cc^2v}$ and $a^{kn}_{cc^2v}$ are pre-computed for each future projection $s(c_i)\in\mathcal{S}(c_i)$. Accordingly, constraints (1.7)-(1.9) ensure that each call in a projection $s(c_i)$ receives an ambulance and that corresponding decision variables for each decision take the correct value based on the ambulance movements.

As part of the implementation, we replace the first call of each scenario in $\mathcal{S}(c_i)$ (hence, the first call of each scenario in each cluster $\mathcal{S}^k(c_i)$) with the current call c_i so that the decision variable corresponding to first call corresponds to the current emergency call that requires an ambulance. In terms of stochastic programming and robust optimization, $x^{kn}_{1c^2v}$ constitutes the first stage decision. Accordingly, as an addition to the D-BLP model, in SBROM we impose the constraint (1.10) that ensures the same ambulance is dispatched for the first call in each future projection.

Two special cases of SBROM are (i) Stochastic-BLP (S-BLP) where the clusters $\mathcal{S}^k(c_i)$ consist of a single future projection (for instance the cluster center) and (ii) Robust-BLP (R-BLP) where we have a single cluster. In case of S-BLP, the cluster center of each cluster is taken as the future projection instead of the worst possible projection. In case of Robust-BLP, the worst member of the single cluster that yields the highest number of late calls is chosen as the future projection. In our experimental results we report performance of these models as well.

In Proposition 1.3, we present an equivalent mixed integer optimization model for solving SBROM. Let $\mathcal{N}(k) := \{1,\ldots,N(k)\}$ denote set of indices for projections in cluster k.

Proposition 1.3. *SBROM is equivalent to the following mixed-integer linear program (SBROM-MILP):*

$$\min_{x^{kn}_{cc^2v}} \sum_{k\in\mathcal{K}} p^k temp_k \tag{1.12}$$

$$s.t.\ temp_k \geq \sum_{c\in C}\sum_{c^2\in C^2}\sum_{v\in B} l^{kn}_{cc^2v}x^{kn}_{cc^2v} \qquad k\in\mathcal{K},\ n\in\mathcal{N}(k) \tag{1.13}$$

$$\sum_{c^2 \in C^2} \sum_{v \in B} x^{kn}_{cc^2v} = 1, \qquad c \in C,\ k \in \mathcal{K},\ n \in \mathcal{N}(k) \tag{1.14}$$

$$\sum_{c^2 \in C^2} x^{kn}_{c^{'}c^2v} + a^{kn}_{cc^{'}v} + \sum_{c^2 \in C^2 c^2 \neq c^{'}} x^{kn}_{cc^2v} \leq 2, \quad v \in B,\ c, c^{'} \in C\ c > c^{'},\ k \in \mathcal{K},\ n \in \mathcal{N}(k) \tag{1.15}$$

$$x^{kn}_{cc^2v} \leq a^{kn}_{cc^2v} \qquad v \in B,\ \ c \in C,\ c^2 \in C^2 c \neq c^2,\ k \in \mathcal{K},\ n \in \mathcal{N}(k) \tag{1.16}$$

$$\sum_{c^2 \in C^2} x^{kn}_{1c^2v} = \sum_{c^2 \in C^2} x^{n^{'}k^{'}}_{1c^2v} \qquad v \in B,\ \ c \in C,\ k, k^{'} \in \mathcal{K},\ n \in \mathcal{N}(k) \tag{1.17}$$

$$x^{kn}_{cc^2v} \in \{0, 1\} \tag{1.18}$$

The SBROM-MILP formulation can be directly solved by any commercial solver, however, as the number of neighbourhood calls N increases, the SBROM-MILP formulation quickly becomes impractical to solve for real life instances. For instance, for a neighborhood of 300 emergency calls ($N = 300$), it takes about 45 seconds for the solution, which is not practical for a real-life decision. In order to bring down the solution time, we employ the algorithm presented in Wang and Jacquillat (2020) for solving scenario based robust optimization problems. We present the results of computational performance in section 1.7. In section 1.5.3 we briefly describe the algorithm and its implementation on our problem. Readers can refer to Wang and Jacquillat (2020) for further details and accompanying proofs.

1.5.3 Solution Algorithm

The scenario based robust optimization problem we are addressing falls under the realm of decision-making under categorical uncertainty. A few examples of this problem include logistics operations under unknown customers with distinct features which are modeled with categorical variables and disaster response under unknown network disruptions where damaged components of the network pose a categorical uncertainty. In our case, consideration of scenarios constitute the categorical uncertainty as scenarios are not expressed with continuous variables but with a series of emergency calls. Accordingly, we leverage the algorithm developed by Wang and Jacquillat (2020) that aims to solve problems with categorical uncertainty. Algorithm 1 provides a step-wise procedure for the proposed algorithm.

At each iteration τ the algorithm considers a subset of scenarios, denoted with $\hat{S}^k_\tau(c)$ for each uncertainty set $S^k(c)$. Then it solves the SBROM-MILP with these subset of scenarios as a master problem, i.e., master problem for the algorithm is the SBROM-MILP with smaller number of future projections than the original problem. The solution to the master problem provides a valid lower bound to the original problem. Note that, the solution to the master problem provides a dispatch decision for the current emergency call c which is the first stage solution. Then, using the first stage solution, an upper bound for the original problem is obtained by evaluating the recourse function for

Algorithm 1 Cutting plane Algorithm for solving SBROM-MILP

Input: $S^k(c)$

Initialization

(a) Set a tolerance $\epsilon = 0.01$ (Optimality gap)

(b) Initialize lower bound $LB = 0$, upper bound $UB = \infty$, and iteration index $\tau = 0$.

(c) Initialize restricted uncertainty sets $\hat{S}^k(c)_1 = \{s_1^k(c)\}$, $\forall k \in \mathcal{K}$

while $(UB - LB)/LB > \epsilon$ **do**

Step 1. $\tau = \tau + 1$ Master Problem:

(a) Solve master problem with subset of scenarios $\hat{S}_\tau^k(c)$

(b) Update lower bound

(c) Retrieve the dispatch ambulance for the current emergency call

(d) Using the dispatch ambulance solve D-BLP model for all scenarios

(e) Update upper bound.

Step 2. Generate new subset of scenarios $\hat{S}^k(c)_{\tau+1}$

(a) Solve the integer program of Equations (A.10)-(A.13).

(b) Update subset of scenarios $\hat{S}^k(c)_{\tau+1}$

end while

Return the dispatch decision

each scenario in the original uncertainty sets. For our problem, the recourse function corresponds to the D-BLP model, i.e., we find the upper bound by solving D-BLP model for each future projections in all of the uncertainty sets given the dispatch decision for the current emergency call. We denote the objective function value of the second stage solution for each future projection (scenario) s in the uncertainty set k at iteration τ with $Z(\tau, s, k)$. Then, upper bound at iteration τ is found by $\sum_{k\in\mathcal{K}} p_k \times \max_{s\in\mathcal{N}(k)} Z(\tau, s, k)$. Once an upper bound and a lower bound is obtained for the current subset of scenarios, the algorithm finds a new subset of scenarios. This step requires a solving an integer optimization problem which we present in Appendix A.1.4. These two steps are repeated until the upper bound and the lower bound converge to each other.

1.6 Experimental Setup

In this section we present the experimental setup with which we compare our methods with that of the benchmarks. Section 1.6.1 describes the series of benchmarks; Section 1.6.2 presents the simulation environment in which we test the different dispatch methods.

1.6.1 Benchmarks

We begin with introducing the benchmark methods. For a fair comparison, we extend the assumptions introduced in Section 1.5 to the benchmark methods. The first method is the Closest Dispatch Policy (CDP), in which the closest available ambulance is dispatched for the emergency call c; the closest ambulance is determined based on the time it will take ambulance v to respond call c, which

is calculated using locations of the emergency call c and ambulance base v where v is chosen among the available ambulances. CDP is the greedy method of dispatching that is most commonly used as a benchmark in the literature (Jagtenberg et al., 2017b).

The second method is Closest Dispatch Policy with Overlaps (CDP-O). This method is a modification of the CDP policy in which a busy ambulance can be queued to respond emergency call c after responding to another call. Similar to CDP, the closest ambulance is determined based on the time it takes for ambulance v to respond call c while taking into account the time it will take for ambulance v to become available (which is 0 for already available ambulances). Based on Assumption 3, the time at which ambulance v becomes available is known in advance and hence calculation of response time of ambulance v to call c is simply time it takes for ambulance v to become available at its base plus time its takes ambulance v to reach the location of the call c. Once respond times of all ambulances are found, ambulance with the smallest response time is dispatched to call c.

The third method is the Dynamic Maximum Expected Covering Location Problem for Dispatching (DMEXCLP-D) heuristic that is first introduced in Jagtenberg et al. (2017a). DMEXCLP-D is the tactical dispatch method that is most close to our method. While there are other methods that consider dispatching in ambulance location and relocation problems, they are not at the same tactical level as we are, and further mix the location-relocation decisions into the decision-making frame. We implement the DMEXCLP-D method and tune its parameters for our data and setting.

The fourth method is D-BLP as it gives the best dispatch policy for a given scenario. Recall that D-BLP provides the upper bound on the performance of any dispatch policy.

The fifth through ninth methods are variations of the SBROM method by which we display the benefits of incorporating uncertainty, Closest Neighbours Clustering and worst-scenario considerations. In the fifth, sixth and seventh methods, we do not employ the worst-scenario selection, i.e., the maximization over the members of the cluster is removed from the formulation. Accordingly, the remaining formulations can be considered to be S-BLPs for which we modify the Closest Neighbours Clustering stages. In the fifth method, we skip the first stage of Closest Neighbours Clustering and instead of finding a set of close calls for the current emergency call c, we use the whole training data set in forming of the scenario sets, i.e., we form scenarios starting with each emergency call in the training data, and cluster these scenarios. In short, in order to make a dispatch for an emergency call c, we solve the S-BLP with the pre-set future projections that were found by clustering of whole training set. By having the fifth method as a benchmark we showcase the benefits of using close calls to the emergency call c, as other calls may dilute the future projection. We refer to fifth method as S-BLP-F. Note that it is practically impossible to employ the worst-projection in the formulation of S-BLP-F, as the size of clusters is too large for computations to finish in reasonable amount of time.

In the sixth method, we exclude the third stage of Closest Neighbours Clustering; instead we find M closest calls to form M future projections with each projection having equal probability. In other words, we skip the clustering step, and set each scenario starting with the close calls to be the future

projection for the optimization step. Notice that due to large computational times, the choice of M is significantly smaller than choice of N. More specifically, we choose M to be close to the choice of K where $M << N$ and $K << N$. By having the sixth method as a benchmark, we showcase the benefits of having a sufficient number of close calls to be able to use clustering on the associated scenarios to obtain a practical solution time to the problem. We refer to this method as S-BLP-M.

In the seventh method, we employ all stages of the Closest Neighbours Clustering and choose the center of each cluster as the representative projection from that cluster. We use this method as a benchmark to showcase the benefits of picking the worst-projection (robustification) when choosing a member of each cluster. We refer to this method as simply Stochastic-BLP (as mentioned earlier) and use S-BLP as its abbreviation.

In the eight method, we use a single cluster of future projections and omit the consideration of different scenarios. By employing this method we highlight the benefits of using multiple clusters of future projections and considering more than a single scenario. We refer to this method as simply Robust-BLP (as mentioned earlier) and use R-BLP as its abbreviation.

In the ninth method we use a single future projection to feed into optimization and do not consider any future uncertainty. We neither consider multiple clusters of future projections nor do the single cluster have multiple members. We simply take the closest call and in the training data and form the future projection based on the series of emergency calls that arrived after this call. In other words, we employ Closest Neighbours Clustering with $N = 1$ and $K = 1$ and skip the clustering step. By employing this method we highlight the benefits of considering multiple future projections and incorporating uncertainty to the decision making. We refer to this method as Online Deterministic BLP and use OD-BLP as its abbreviation.

1.6.2 Simulation

In this section we present the simulation procedure with which we test the different dispatch methods. Algorithm 2 displays a detailed pseuo-code of the simulation procedure. For all dispatch methods except D-BLP, we employ Algorithm 2 to obtain the dispatch policy and the accompanying performance measures for a given real-life scenario s. For D-BLP we first compute the required inputs l_{cc^2v} and a_{cc^2v} and feed these inputs to obtain the dispatch policy for scenario s. We keep track of two important performance measures, (i) response time for each emergency call; and (ii) whether if ambulance was late to response, i.e., if the emergency call received the ambulance later than the threshold response time. Using the response times, we compute the average response time and average response time for late calls, and using the late responses we compute the total number of late responses for an experiment. Note that the objective functions of all optimization methods aim to minimize the number of late responses, but also has an impact on the average response time as well as average response time for late calls.

The inputs to the simulation procedure are *postTime*, *timeThreshold*, a real-life scenario

Algorithm 2 Simulation for testing

Input: $postTime, timeThreshold, s = \{c_1, c_2, ..., c_s\}$
Input: $travelTimeMatrix[c, v]$, $dispatchMethod(\dots)$, $conditionalClutering(\dots)$
  $availabilityOfAmbulances[\mathbb{B}] \leftarrow \mathbb{1}$
  $availableTimesOfAmbulances[\mathbb{B}] \leftarrow \mathbb{0}$
  $responseTime[s] \leftarrow \mathbb{0}$
  $dispatchedAmbulance[s] \leftarrow \mathbb{0}$
  $penalty[s] \leftarrow \mathbb{0}$
  $currentCall \leftarrow 0$
  $currentTime \leftarrow 0$
  while $currentCall \in s$ **do**
    $currentTime \leftarrow currentCall[t]$
    for all $v \in \mathbb{B}$ **do**
      if $availableTimesOfAmbulances[v] < currentTime$ **then**
        $availableTimesOfAmbulances[v] \leftarrow 0$
        $availabilityOfAmbulances[v] \leftarrow 1$
      end if
    end for
    $[clusters, probabilities] \leftarrow conditionalClutering(currentCall, \dots,$
    $\dots, availabilityOfAmbulances, availableTimesOfAmbulances)$
    $dispAmb \leftarrow dispatchMethod(clusters, probabilities)$
    $dispatchedAmbulance[currentCall] \leftarrow dispAmb$
    if $availabilityOfAmbulances[dispAmb] == 1$ **then**
      $responsTime[currentCall] \leftarrow travelTimeMatrix[currentCall, dispAmb]$
    else
      $responsTime[currentCall] \leftarrow availableTimesOfAmbulances[dispAmb] - currentTime +$
      $travelTimeMatrix[currentCall, dispAmb]$
    end if
    $availabilityOfAmbulances[dispAmb] \leftarrow 0$
    $availableTimesOfAmbulances[dispAmb] \leftarrow travelTimeMatrix[currentCall, dispAmb] +$
    $postTime$
    if $responsTime[currentCall] < timeThreshold$ **then**
      $penalty[currentCall] \leftarrow 1$
    else
      $penalty[currentCall] \leftarrow 0$
    end if
  end while

$s = \{c_1, c_2, \ldots, c_s\}$, $travelTimeMatrix[c, v]$, $conditionalClustering(\ldots)$ and $dispatchMethod(\ldots)$. The input $postTime$ is the simulation parameter for how much time an ambulance becomes busy after arriving to the scene of an emergency call. As noted in **Assumption 3**, this parameter is known and deterministic. We set its value to be 20 minutes for our experiments. The input $timeThreshold$ is the time threshold for an ambulance response to be considered late. For our main results, we use the $timeThreshold = 9$ minutes below.

For testing purposes, we utilize the test data set we have introduced in section 1.3. The real-life scenarios we consider in experiments consist of emergency calls in a day (24-hour period) starting from 4 am. We start a day at 4 am because it is time of day with the least amount of activity. This allows us to set the state of all ambulances to be available at the beginning of a scenario. We note that the number of calls in each scenario might differ, as these are real life instances. The input $travelTimeMatrix[c, v]$ is the matrix of travel time for ambulance v to reach emergency call c. For a given scenario s we pre-compute the travel times between each emergency call and ambulance. Based on Assumption 2, this computation is trivial.

The inputs $conditionalClustering(\ldots)$ and $dispatchMethod(\ldots)$ take various forms as described in Section 1.6.1. For instance, for CDP method, the Closest Neighbours Clustering step is not needed and an ambulance is dispatched based solely on the response times.

1.7 Experimental Results

In this section we present the results of our experiments. In Section 1.7.1 we report our main results comparing the performance of our methods against the benchmarks for a series of performance metrics. In Section 1.7.2 we display the sensitivity of our method to the parameters of the Closest Neighbours Clustering step. In Section 1.7.3 we test the robustness of our method to parameters of the experimental setting. Finally, Section 1.7.4 presents the comparison of computational times needed for direct implementation and the algorithm for scenario based robust optimization developed by Wang and Jacquillat (2020).

1.7.1 Performance Comparisons

The benchmarks methods such as CDP, CDP-O and D-BLP, are non-parametric dispatch methods and hence do not require parameter tuning, however, our methods and DMEXCLP-D require parameter tuning. For this purpose, we split the test set $\mathcal{P}^{Ts}$ into two subsets $\mathcal{P}_1^{Ts}$ and $\mathcal{P}_2^{Ts}$. We stress that both subsets have no intersection with the training set used for the Closest Neighbours Clustering: $\mathcal{P}_1^{Ts} \cap \mathcal{P}^{Tr} = \emptyset$ & $\mathcal{P}_2^{Ts} \cap \mathcal{P}^{Tr} = \emptyset$. We use the first subset $\mathcal{P}_1^{Ts}$ to find the best parameters for our methods and use the second subset $\mathcal{P}_2^{Ts}$ to compare the performances of our methods. We report the results for the both subsets. The first subset consist of forty days of experiments with a total 4552 emergency calls and the second subset consists of a hundred and forty days of experiments

with a total of 19126 emergency calls. While the objective of the optimization methods is to minimize the number of late calls, we report the performances of all methods with respect to three different important metrics (i) number of late calls, (ii) mean response time and (iii) mean response time for late calls. Late calls refer to the emergency calls that receive an ambulance after a time threshold. As noted earlier, a single experiments consists of emergency calls in a 24-hour period starting from 4 am and we set the time threshold to be 9 minutes based on Lamb et al. (2014).

Table 1.1 below shows the performances of all methods for the three metrics for both subsets of test data. Results of the first and second subsets are shown under “In-Sample" and “Out-Of-Sample" columns respectively. For the In-Sample column, results are shown for the parameter that performs best in terms of number of late calls.

Methods	Number of Late Calls		Mean Response Time		Mean Response Time of Late Calls	
	I-S	O-o-S	I-S	O-o-S	I-S	O-o-S
CDP	**81**	**240**	3.24	3.27	10.71	10.51
CDP-O	**64**	**195**	3.2	3.22	10.61	10.64
DMEXCLP-D	**44**	**124**	5.5	5.58	14.43	14.67
S-BLP-F	**40**	**97**	4.97	4.89	20.83	23.26
S-BLP-M	**41**	**99**	5.04	4.97	21.36	22.99
S-BLP	**38**	**114**	5.03	4.96	20.26	22.05
R-BLP	**116**	**378**	5.61	5.63	19.87	19.54
OD-BLP	**104**	**325**	5.54	5.58	19.51	19.98
SBROM	**33**	**91**	5.15	5.15	19.05	22.21
D-BLP	**29**	**85**	5.09	4.8	16.25	17.39

Table 1.1: Performances of all methods. I-S stands for In-Sample and O-o-S stands for Out-of-Sample

Observe that the performance of the D-BLP method for both subsets are 29 and 85, respectively. These numbers constitute the upper bound of performance for late calls for the two subsets. With respect to these upper bounds, the performances of CDP and CDP-O methods perform poorly, with a total of 321 and 259 late calls, respectively. One observation we derive from examining the number of late calls under the CDP and CDP-O policies is the potential benefits of allowing future-looking decisions in dispatching. By allowing overlaps in response time frames, performance of the CDP policy was improved by around 19%. In practical terms, if dispatchers also consider ambulances that will become available soon, the system performance in terms of late calls can be significantly improved. We also observe that the performances of CDP and CDP-O methods compared to the D-BLP method for mean response time and mean response time of late calls are over-performing with 3.24-3.27, 3.2-3.22 and 5.15 minutes, respectively. This is an expected result as the CDP and

CDP-O methods myopically send the closest vehicle to an emergency call. Yet, this difference also highlights the well known trade-off between the two objectives of minimizing the number of late calls and minimizing the mean response time (Jagtenberg et al., 2017a). The D-BLP method sacrifices its performance in mean response time to achieve better performance in number of late calls.

The DMEXCLP-D method requires a single parameter q, that indicates the pre-determined probability of an ambulance being unavailable. Accordingly, we estimate the parameter q by computing the total workload of the past emergency calls and dividing it by total number of ambulances. This value approximately equals to 0.18. We experimented (using In-Sample subset) with parameter values between 0.1 and 0.3 and saw that the performance of DMEXCLP-D is unaffacted by small perturbations around 0.18. While the DMEXCLP-D method performs noticeably better than the CDP and CDP-O methods as noted by Jagtenberg et al. (2017b), there is still room for improvement as it yields 44 and 124 late calls in the In-Sample and Out-of-Sample subsets, respectively. Moreover, its performance in mean response time is inferior to the D-BLP method. This is an important observation as the D-BLP method achieves the minimum number of late calls by sacrificing the mean response time compared to CDP and CDP-O policies. However, the DMEXCLP-D method performs worse than the D-BLP in terms of mean response time as well.

Among online methods we note that SBROM performs the best with 33 and 91 late calls in In-Sample and Out-Of-Sample subsets. Its outcomes achieve 59%-62%, 48%-43% and 25%-26% performance improvements over CDP, CDP-O and DMEXCLP-D methods. Moreover, we observe that SBROM achieves a smaller number of late calls compared to the DMEXCLP-D method while also maintaining smaller mean response times. Given the trade-off between the mean response time and number of late calls, our method, SBROM, achieves improvement over both metrics when compared to state of the art methods in the literature. These improvements highlight the benefit of using the SBROM method for ambulance dispatching. While these improvements are worthwile, one setback of the SBROM method is its performance in mean response time for late calls: In achieving a smaller number of late calls, the SBROM methods sacrifices some performance in the response time of late calls.

Next we compare the performance of the SBROM method with its variants (S-BLP-F, S-BLP-M, S-BLP, R-BLP and OD-BLP) to understand the benefits of Closest Neighbours Clustering as well as Stochastic Programming and Robust Optimization. First and foremost, we observe that the OD-BLP method performs very poorly with 104-325 late calls (higher than even the CDP method) in In-Sample and Out-Of-Sample subsets. The SBROM method outperforms OD-BLP method by around 68%. Moreover, the other methods S-BLP-F, S-BLP-M, S-BLP outperform the OD-BLP method by approximately 61%. These differences highlight the importance of considering future uncertainties by employing multiple future projections.

In a similar spirit, we observe that the R-BLP method performs even worse than the OD-BLP method with 116 and 378 late calls. This further highlights the benefits of considering multiple projections in decision making as well as the poor performance of picking a single worst case among

available scenarios. Accordingly, when multiple scenarios are considered in the decision making we observe the performance improvements at S-BLP-F, S-BLP-M and S-BLP methods where they outperform the DMEXCLP-D, CDP and CDP-O methods with 40-97 41-99 and 38-114 late calls, respectively.

When comparing the performances of S-BLP-F, S-BLP-M and S-BLP, we observe that S-BLP achieves the best In-Sample performance with 38 late calls but falls behind both S-BLP-F and S-BLP-M methods with 114 late calls in the Out-Of-Sample performance. Based on the In-Sample performance of S-BLP method, we see that all three stages of the Closest Neighbours Clustering Step are useful in improving performance. However, based on the Out-Of-Sample performance, we conclude that the choice of parameters related to stages 1 and 3 are key for good dispatch decisions. (We examine the sensitivity to parameters of Closest Neighbours Clustering for the SBROM method in section 1.7.2).

Lastly, the SBROM method outperforms S-BLP-F, S-BLP-M, S-BLP and R-BLP methods by around 18%-7%, 20%-9%, 16%-20% and 67%-73% at In-Sample and Out-Of-Sample subsets, respectively. Together, these differences highlight the benefit of employing all three stages of Closest Neighbours Clustering along with combining Stochastic Programming and Robust Optimization - considering a sample of future projections by utilizing clusters of scenarios and protecting the dispatch decisions against adversarial situations around the existing scenarios. We stress that the SBROM method utilizes the clusters as uncertainty sets and weights multiple scenarios that are chosen adversely from each of the clusters. In doing so it exploits the structure of the data provided by the Closest Neighbours Clustering. In other words, SBROM model and Closest Neighbours Clustering steps complement each other in making of the best online dispatch decision.

1.7.2 Sensitivity to Parameters

In Fig. 1.1 we show the sensitivity of SBROM method to two important parameters of the Closest Neighbours Clustering step: (i) sample size N and (ii) number of clusters N. Sample size N determines the number of scenarios visited as part of the optimization and number of clusters determine how many scenarios to choose when taking the expectation for the future projections. Fig. 1.1 also displays the parameter-tuning of the SBROM method, namely, we show the performance of SBROM method under various parameter settings for the first subset of the test data set.

Our observations are twofold. First, the best performance of the SBROM method is achieved with an intermediary sample size. By increasing the sample size from $N = 100$, the performance of SBROM method improves until a certain point of N. At sample size $N = 400$, we see that the performance of SBROM method deteriorates compared to $N = 300$ except for 2 instances. This phenomenon is related to well-known bias-variance trade-off in machine learning literature. When a small N is chosen, possibly informative future projections are excluded from the decision making and hence dispatch decision is not prepared for some highly possible situations. On the other hand,

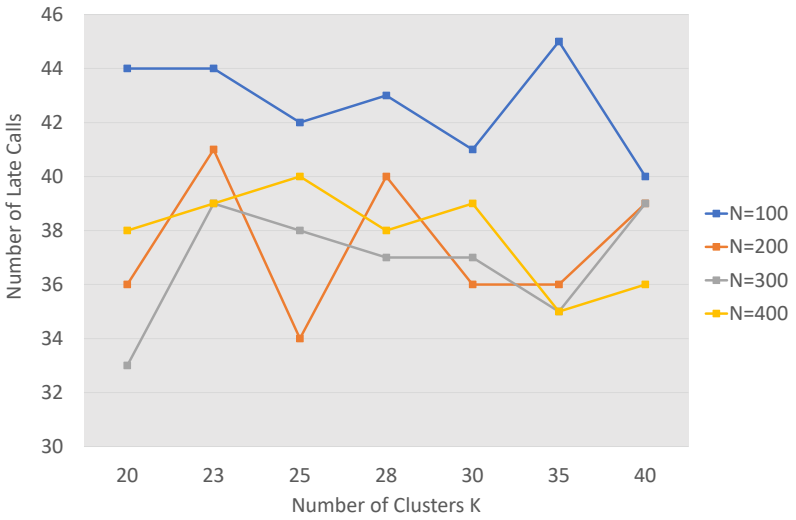


Figure 1.1: Sensitivity of number of late calls to number of clusters K and number of neighbours N

when N is large, the dispatch decision becomes too conservative for realistic situations as there are possibly more adverse scenarios in large uncertainty sets.

Second, performance of the SBROM method fluctuates non-monotonically as the number of clusters K increases. For a sample size $N = 300$, the performance of SBROM method fluctuates around 36 late calls. While this is a notable value, we note all the experiments yield better performance than state-of-the art methods (which had 44 late calls for the same set). Moreover, based on this, unlike the S-BLP method, we see that SBROM method is more robust to the changes in the parameters.

1.7.3 Robustness to Experimental Setting

Our results so far showcase the benefits of our SBROM method in comparison with state-of-the art method DMEXCLP-D as well as variations of the SBROM under a specific experimental setting (where $timeThreshold = 9$ and $postTime = 20$). We now establish the the robustness of SBROM method under different experimental settings. By changing $timeThreshold$, we simply make changes to the output of the pre-computation step by modification of the penalty term l_{cc^2v}. Similarly, by changing $postTime$, we simply make changes to the output of the pre-computation step by modification of the indicator term a_{cc^2v}. For all the experiments in this section, we use the best parameters we have found earlier for all the methods. We utilize the first subset with forty days of experiments.

Table 1.2a shows the robustness of SBROM method to different $timeThreshold$ values. First, note that a decrease (increase) in $timeThreshold$ makes the dispatching problem more (less) chal-

	timeThreshold (min)						
	4	5	6	7	8	9	10
CDP	1261	773	465	257	148	81	43
CDP-O	1218	739	446	240	131	64	33
DMEXCLP-D	1418	790	448	213	107	44	17
S-BLP-F	1246	767	449	206	92	40	15
S-BLP-M	1242	738	429	212	102	41	19
S-BLP	1224	746	428	208	94	38	22
R-BLP	1547	941	597	374	237	116	61
OD-BLP	1486	905	569	321	195	104	55
SBROM	1148	696	416	201	88	33	15
D-BLP	1117	650	389	186	80	29	9

(a) *timeThreshold*

	postTime (min)				
	10	15	20	25	30
CDP	45	58	81	100	122
CDP-O	35	46	64	88	109
DMEXCLP-D	28	35	44	49	71
S-BLP-F	24	34	40	54	59
S-BLP-M	24	33	41	57	66
S-BLP	26	34	38	56	65
R-BLP	63	87	116	133	190
OD-BLP	56	77	104	132	179
SBROM	24	32	33	47	54
D-BLP	23	28	29	29	37

(b) *postTime*

Table 1.2: Performance of all methods under different *timeThreshold* and *postTime* values.

lenging and accordingly, the number of late calls increase (decrease) for any given method as *timeThreshold* increases. Second, the SBROM method out-performs all other online methods for all the *timeThreshold* values. An important observation about $timeThreshold = 4$ column is that the CDP and CDP-O methods out-perform the DMEXCLP-D as well as variations of SBROM method. This highlights the difficulty of dispatching problem when the *timeThreshold* is very small. Furthermore, the fact that SBROM method out-performs the CDP and CDP-O methods even when $timeThreshold = 4$, emphasize the benefit of the SBROM method.

Table 1.2b shows the robustness of SBROM method to different *postTime* values. First, note that an increase (decrease) in the *postTime* makes the dispatching problem more (less) challenging and accordingly, the number of late calls increases (decreases) for any any given method as *postTime* increases. Second, the SBROM method out-performs all other online methods again for all the *postTime* values. An important observation about $postTime = 10$ column is that the performances of SBROM method and its variants S-BLP-F, S-BLP-M and S-BLP are very close to the performance of the D-BLP. This is because, when ambulances take less time to become available, dispatching problem becomes easier. However, as *postTime* increases, the performance of the SBROM method over its variants improve and stay closer to the D-BLP. This highlights the benefits of the SBROM method and showcases its performance even when difficulty of dispatching problem increases.

1.7.4 Computational Time

We conclude the experimental results with the performance of the algorithm we borrowed from Wang and Jacquillat (2020). While we did not develop the algorithm for this chapter, we find the discussion of the computational time to be a relevant topic for producing timely dispatch decisions

as we aim to make a dispatch decision in a matter of seconds.

Table 1.3 reports the test results for a single scenario with 115 emergency calls, i.e, both the algorithm and direct implementation of SBROM-MILP formulation is run 115 times to produce 115 dispatch decisions. We tested for various problem sizes by changing both the number of clusters K (Column 1) and sample size N (Column 2). Column 3 reports the average computational time with Gurobi implementation of the SBROM-MILP formulation. The next four columns show whether if the algorithm converged, the computational time, number of iterations and number of scenarios considered for the algorithm.

		Gurobi		Algorithm Performance		
Number of Clusters	Sample Size	CPU (s)	(UB-LB)/LB	CPU (s)	Number of Iterations	Number of Scenarios Considered
1	50	**1.55**	0	**1.08**	1.97	2.53
1	100	**5.42**	0	**2.67**	2.14	2.62
1	200	**20.25**	0	**4.75**	2.19	2.59
1	300	**43.12**	0	**8.79**	2.2	2.7
5	50	**1.55**	0	**1.23**	1.97	9.93
5	100	**5.42**	0	**2.71**	2.24	11.82
5	200	**21.47**	0	**5.61**	2.43	12.51
5	300	**43.44**	0	**8.35**	2.54	13.06
10	50	**1.56**	0	**1.44**	1.99	17.04
10	100	**5.32**	0	**2.93**	2.16	20.85
10	200	**20.71**	0	**5.82**	2.4	23.56
10	300	**43.62**	0	**8.16**	2.43	24.15
20	50	**1.56**	0	**1.76**	1.83	28.98
20	100	**5.34**	0	**3.49**	2.15	36.21
20	200	**19.75**	0	**6.6**	2.35	43.36
20	300	**45.86**	0	**9.15**	2.52	45.96
30	50	**1.56**	0	**2.08**	1.66	39.69
30	100	**5.47**	0	**4.15**	2.09	48.73
30	200	**18.48**	0	**6.75**	2.4	62.06
30	300	**42.21**	0	**9.93**	2.49	65.96
50	50	**1.56**	0	**2.1**	1	50
50	100	**5.41**	0	**5.58**	1.87	69.27
50	200	**18.5**	0	**8.68**	2.32	89.04
50	300	**42.48**	0	**11.53**	2.41	100.15

Table 1.3: Computational Performance of the Algorithm by Wang and Jacquillat (2020)

First, based on the convergence column, the algorithm converges to the optimal solution in all instances of the the dispatching problem. Second, as the sample size N increases the computational time for both direct implementation and algorithm increase. However, the increase in direct implementation is significantly larger than that of algorithm. Specifically, for a setting that results in quality results which is highlighted in Table 1.3, the computational time for direct implementation takes around 45 seconds for a single decision while the it is 9.15 seconds for the algorithm. The algorithm drops the computational time of dispatch problem by 80%. Given the need for rapid decision making, the algorithm of Wang and Jacquillat (2020) performs very well for real-life dispatching circumstances. Third, the algorithm remains stable when the sample size and number of clusters increase, i.e., the average number of iterations required by the algorithm increase by 1-1.5 while the sample size and cluster size increase by 6-fold and 50-fold.

1.8 Conclusion

In this chapter we study the problem of ambulance dispatch in Emergency Medical Services with a novel data-driven methodology that combines ideas from machine learning and decision making under uncertainty. Our method extends the operations literature on dispatching of emergency medical services and sets a new benchmark for dispatching problem. In collaboration with University of Pittsburgh Department of Emergency Medicine and the City of Pittsburgh Bureau of EMS we test our methodology on unique data set from city of Pittsburgh, Pennsylvania.

As part of the solution methodology, we first propose a omniscient deterministic binary linear program that incorporates forward looking dispatch decisions. Using the deterministic model as a basis we build a scenario based robust optimization formulation that produces dispatch decisions under future uncertainty. Scenario based robust optimization is a novel modelling approach that combines ideas from stochastic programming and robust optimization by constructing probabilistic scenario sets and protecting against adversarial scenarios within the scenario sets. To this end, we also develop the Closest Neighbours Clustering method, a data-driven methodology for building the scenario sets from real data. We show the benefits of using scenario based robust optimization along with Closest Neighbours Clustering against a series of benchmarks including stochastic programming and robust optimization. Moreover, our methodology outperforms the state-of-the art dispatching methods in the existing literature by a significant margin.

This chapter successfully develops and uses the Closest Neighbours Clustering and scenario based robust optimization on the dispatching problem with real data. Our results for this unique data set are significant and promising for application of our methodology on new data sets. It would be an interesting future research direction to employ these methods on a different context. For instance, investment decisions in financial markets is an interesting data-rich context on which these methods could be utilized.

Chapter 2

Analysis and Comparison of Two-Sided Queues with Different Levels of Delay Information

2.1 Introduction

A service firm can disclose delay information to its customers to induce desirable joining/balking behavior. A considerable amount of research has studied delay information disclosure policies in traditional *one-sided* markets where a firm provides service to its users (i.e., customers); see Ibrahim (2018) for a review. With the rise of so-called "on-demand platforms", which are typically *two-sided markets*, the decision of sharing delay information with users has become richer in its scope. On-demand platforms facilitate matches between two *user classes*: *Consumers*, who seek service through the platform, and *providers*, who supply the service to the consumers through the platform. Examples of on-demand platforms include ride-sharing services (e.g., Uber and Lyft), on-demand food and grocery delivery (e.g., GrubHub, Doordash, and Instacart), and labor marketplaces (e.g., TaskRabbit). In this chapter, we model the operations of an on-demand platform as a two-sided queue and we investigate the platform's optimal delay information disclosure policy when users on both sides of the market are delay-sensitive. This optimal policy seeks to maximize the average rate at which the platform successfully matches consumers with providers (*match rate*); we use the match rate as a proxy for the platform's revenue and profitability.

An important aspect of on-demand platforms is the voluntary nature of users' participation. The providers are not employees of the platform but rather independent contractors who make their own decisions regarding when they work, when they take a break, and when they call it a day. Consumers are also often not loyal to a specific platform and can easily switch to other platforms to receive service for various reasons, including long delays to be matched with a provider. In response to

this challenge of "at-will participation," platforms have employed a variety of incentive structures to influence users' participation decisions to maximize profitability. For example, through *surge pricing* mechanisms, platforms vary their service price to balance the supply and demand. We study another mechanism that influences participation decisions, which is the type of delay information that the platform shares with consumers and/or providers.

It is not immediately clear *how best* a platform can use the leverage of sharing delay information. Indeed, firms have been indecisive about such decisions in the past: for example, Uber, for a brief period, revoked its drivers' ability to see how many other drivers were in the vicinity (UberPeople.net, 2017). This decision was hotly contested by the drivers as it diminished their ability to estimate how long they would have to wait to receive a ride request. Furthermore, there are various *types* of information to disclose. For example, ride-sharing platforms provide riders with a current estimate of their wait time, and provide drivers with coarser information through heat maps, where different colors indicate different levels of anticipated delay.

In this chapter, we focus on three information regimes—the *occupancy information* where *both* consumers and providers are informed of the current occupancy level in the platform (equivalently, the expected delay conditioned on the current number of users in the platform), and two *asymmetric* information regimes where *either* consumers or providers are provided with occupancy information, while the other side is provided with no information. We study how the choice of information regime affects the platform's profitability by analyzing and comparing the match rates resulting from employing each of these regimes. In our model, users obtain a reward for receiving service and incur a delay cost. Arriving users form an expectation of the delay cost they will incur based on the delay information provided to them by the platform. Based on this expectation, they decide whether to join the platform or balk in order to maximize their utility.

Below, we summarize the main contributions of this chapter:

- We model an on-demand platform as a two-sided queuing system that facilitates matches between two user classes. Users in each class are heterogeneous in their delay sensitivities. We contribute to the literature by studying the platform's information regime choice, that determines what information to disclose to users of *both* user classes.

- We compute the effective match rates associated with each information regime. We characterize sufficient conditions for each regime to emerge in equilibrium and provide insight into the conditions under which the platform prefers to hide information from one user class. Specifically, we examine the optimal information regime for two limiting settings. First, we examine a setting where the two user classes are highly imbalanced in their patience profiles. Second, we examine a setting where users of one class are relatively abundant. We identify the conditions for the platform to prefer sharing no delay information to one of its user classes in both these limiting settings.

- In an extension, we demonstrate that for systems with higher levels of user heterogeneity (a

larger number of distinct patience sensitivity levels), the platform prefers to provide occupancy information to at least one of its user classes for a wider range of parameters—this complements the results of Guo and Zipkin (2007) and Dobson and Pinker (2006) for one-sided queuing systems.

2.2 Literature Review

We contribute to the recent literature on on-demand platforms. Wang and Yang (2019) survey literature on different aspects of on-demand platforms. The aspects of this literature that have received the most attention in the Operations Management literature are (i) pricing (e.g., Cachon et al., 2017; Taylor, 2018; Hu and Zhou, 2019); (ii) matching (e.g., Dickerson et al., 2018; Lyu et al., 2019; Özkan and Ward, 2020); and (iii) the impact of sharing different types of information with the users of such platforms. The existing papers on information sharing mostly focus on (i) disclosing *fare* or *destination* information to drivers in ride-sharing platforms (e.g., Rosenblat and Stark, 2016; Chu et al., 2018) and (ii) sharing *customers' attributes* with providers (e.g., Romanyuk, 2017; Romanyuk and Smolin, 2019); they all conclude that, under some conditions, full disclosure could hurt the platforms' performance. Unlike the mentioned information sharing papers, our focus is on sharing *delay information*. Furthermore, we consider the implications of sharing such delay information with *both* classes of users (consumers and providers). Delay information sharing has received extensive attention in traditional one-sided queues, but has not yet received much attention in the two-sided settings.

Hassin (2016) provides a comprehensive review of literature on one-sided queuing systems in the presence of strategic customers, who make joining decisions using the delay information available to them (strategic queues). Ibrahim (2018) provides a comprehensive review of literature on delay announcement in strategic queues, some of which analyze the accuracy of various types of delay information structures (Armony et al., 2009; Ibrahim et al., 2017) and some of which empirically study the impact of delay information on users' behavior (Akşin et al., 2013; Batt and Terwiesch, 2015; Yu et al., 2017). The stream of delay announcement literature which most closely relates to our work studies how various delay announcement structures can improve system performance. For instance, Hu et al. (2018) consider the impact of informing only a fraction of customers about real-time delay in a single-server queue, and find that some amount of information heterogeneity can increase both throughput and social welfare. Dimitrakopoulos et al. (2019) study a single-server system in which the queue visibility alternates between 'observable' and 'unobservable' periods, and find that setting the duration of these periods optimally improves, in general, the equilibrium throughput and social welfare, compared to a purely observable or a purely unobservable queue. Lingenbrink and Iyer (2019) study the structure of the optimal information disclosure policy in a single-server queue where the strategic customers estimate their expected delay based on Bayesian updating. They find that the throughput-optimal delay signalling mechanism is of threshold type,

i.e., the platform's delay signal must induce customers to join only up to a certain occupancy level.

The papers that study delay announcement structures most similar to ours are Guo and Zipkin (2007) and Dobson and Pinker (2006). Like ours, both papers analyze and compare different richness levels of the provided delay information, however, in a one-sided setting. In a single-server queue, Guo and Zipkin (2007) employ a general model of customer patience heterogeneity, and find that when customers are sufficiently heterogeneous, disclosing richer information leads to higher throughput. Similarly, Dobson and Pinker (2006) consider sharing lead-time information in a supply chain and show that when users are more heterogeneous in their patience levels, sharing richer lead-time information improves throughput. We find that this result extends to a two-sided setting: By comparing the throughput (which in our case is the effective match rate) under three specific patience heterogeneity distributions, we show that sharing richer delay information benefits the platform if the users are more heterogeneous. Similar to the results of Guo and Zipkin (2007) for a one-sided queue, we find that, in a two-sided setting, the platform's optimal information structure *can* be in the best interest of the users.

2.3 Model

We consider a two-sided queuing system (the platform) with two classes of *users*—consumers and providers—where consumers arrive to one side of the system and providers arrive to the other side. Consumers use the services/products offered on the platform, while providers supply these services/products. A match occurs instantly if a consumer (resp., provider) arrives while a provider (resp., consumer) is waiting to be matched; waiting users are cleared based on the first-come-first-served (FCFS) discipline. Matched user pairs leave the system instantly. Therefore, the system never has both consumers and providers waiting to be matched at any given time. Users of both classes are delay-sensitive and seek to maximize their utility (we define the utility functions in Section 2.3.2). The platform may decide to relay delay information to one or both user classes, based on which arriving users form an expectation of their system delay (time to match) and accordingly make irreversible join/balk decisions. The types of relayed delay information form an information regime, I, which induces a mapping from the current *system state* (the number and class of users waiting to be matched) to *delay signals* provided to each user class. In Section 2.3.1, we describe the information regimes that we consider in the chapter. Subsequently, in Section 2.3.2 we introduce our model for users and explain how they make joining/balking decisions.

2.3.1 Information Regimes

The platform's manager chooses an information regime that maximizes the *effective match rate* (i.e., the rate at which users are matched and leave the system), which can be considered as a proxy for revenue in on-demand platforms. Based on the "No Information" and "Partial Information" regimes

studied for the one-sided queuing setting in Guo and Zipkin (2007), we construct and study the following information regimes for our two-sided setting:

- **Occupancy information (Regime $\mathbb{O}$):** In this regime, the platform signals to arriving users of both classes the current system state (which can be, equivalently, considered as signaling the current queue-length based expected delay under the FCFS discipline).
- **Occupancy information to consumers only (Regime $\mathbb{C}$):** In this regime, the platform relays the current system state to consumers, and relays no information to providers.
- **Occupancy information to providers only (Regime $\mathbb{P}$):** In this regime, the platform relays the current system state to providers, and relays no information to consumers.

When no information is provided to either user class (the fourth possible combination), users of each class arrive at fixed rates (independent of the system state). Since the arrival rate of one user class functions as the service rate for the other user class in a two-sided queue, the resulting system cannot be stable, and for this reason, we disregard this information regime.

2.3.2 Users' Joining Decisions

We denote a user by superscript $u \in \{c, p\}$ where c and p stand for consumers and providers, respectively. Consumers and providers arrive to their side of the platform with independent Poisson rates Λ^c and Λ^p, respectively. Users receive zero utility from balking, and join only if doing so yields a positive expected utility. Joining decisions are irreversible, i.e., users who join wait till they are matched (no abandonment). A user's utility depends on her valuation for a match ($R^u \geq 0$) and her expected delay cost, which itself depends on:

1. The user's *individually drawn* delay sensitivity θ^u, which is a realization of the random variable $\Theta^u \sim f^u(\theta^u)$; the probability distribution $f^u(\theta^u)$ captures the heterogeneous delay sensitivity of users of the same class, and it has a finite support $[0, 1]$.
2. The expectation of a *common cost function* $c^u(\cdot)$ of the delay W_I^u that the user will experience; this expectation is based on the signal s that the user observes and the information Regime I. We denote this expectation by $\mathbb{E}[c^u(W_I^u(s))]$, and for tractability, we consider a linear function for it, i.e., $\mathbb{E}[c^u(W_I^u(s))] = a^u\,\mathbb{E}[W_I^u(s)] + b^u$, where a^u and b^u parametrize this cost function.

Therefore, the expected utility of user u with delay sensitivity θ^u who receives signal s under information Regime I follows:

$$U_I^u(s) = R^u - \theta^u \left(a^u\,\mathbb{E}[W_I^u(s)] + b^u\right), \quad u \in \{c, p\}. \tag{2.1}$$

We set $b^u = R^u$ to ensure that $U_I^u(s) \geq 0$ when $\mathbb{E}[W_I^u(s)] = 0$ and $U_I^u(s) < 0$ when $\mathbb{E}[W_I^u(s)] > 0$ and $\theta^u = 1$; this excludes those users who would balk even when the signal indicates no delay,

and ensures that the most delay sensitive users ($\theta^u = 1$) balk when the signal indicates delay. For convenience and without loss of generality, we scale time by a multiplicative factor of a^c/R^c, and then to leave the utility function unchanged, we scale a^u by a factor of R^c/a^c. The resulting utility functions follow:

$$\begin{aligned} U_I^c(s) &= R^c\left(1-\theta^c\left(1+\mathbb{E}[W_I^c(s)]\right)\right), \\ U_I^p(s) &= R^p\left(1-\theta^p\left(1+K^p\,\mathbb{E}[W_I^p(s)]\right)\right), \end{aligned} \tag{2.2}$$

where $K^p = (a^p/a^c)(R^p/R^c)$ represents the relative value of time for the providers compared to the consumers; i.e., for the same R^u and θ^u values, if $K^p > 1$ then providers value time more than consumers do, and therefore, they are willing to wait less to be matched than consumers are.

We assume that users can compute their expected delay (a typical assumption in the strategic queuing literature; for example, see unobservable queueing models in Hassin, 2016) based on the signal that they observe. This computation, in turn, depends on the equilibrium joining behavior of users of both classes. We formalize this behavior by defining an equilibrium joining strategy as a set of probabilities, $\{J_I(\theta^u|s); \forall \theta^u, s\}$, each specifying the probability that a user of class u with delay sensitivity θ^u joins upon observing the signal s under information Regime I. For this set of joining probabilities to form an equilibrium, it must be that every focal user has no incentive to deviate from the equilibrium joining probability $J_I(\theta^u|s)$, keeping all other equilibrium joining probabilities fixed.

We now turn to the issue of modeling the heterogeneity in users' delay sensitivities using $f^u(\theta^u)$. For tractability and clarity of insights, our base model considers two user *types* within each class: patient and impatient; patient users are willing to wait for a match, while impatient users are not. Formally,

$$f^u(\theta^u) = \begin{cases} \delta^u & \text{for } \theta^u = t^u < 1 \\ 1-\delta^u & \text{for } \theta^u = 1 \end{cases}, \qquad u \in \{c, p\}.$$

Under this two-point user heterogeneity model, *impatient* users (for whom $\theta^u = 1$) arrive at rate $\Lambda^u(1-\delta^u)$ and *patient* users (for whom $\theta^u = t^u < 1$) arrive at rate $\Lambda^u\delta^u$. Under this distribution, impatient users join with a positive probability, in equilibrium, if and only if they expect a delay of zero. On the other hand, patient users may join with positive probability in equilibrium even if they expect a positive delay (see Eq. (2.2)). This two-point distribution allows us to tractably capture the impact of user heterogeneity on the platform's information regime choice. We consider extensions where one or both user classes have delay sensitivities that follow a continuous uniform distribution in Section 2.6.

2.4 Analyzing the Information Regimes

As discussed earlier, the choice of information regime influences the equilibrium joining strategies of consumers and providers, and consequently, the platform's effective match rate. As a step towards

obtaining the effective match rates under each regime, we first construct a general underlying Continuous Time Markov Chain (CTMC) that we use later to represent and analyze the dynamics of the system under each information regime. As matches occur instantaneously, there are either no users, or providers, or consumers (but never both providers and consumers) in the system at any point. Therefore, the CTMC of the system under any of the information regimes has a one-dimensional state space with the state variable $n \in \mathbb{N} = \{\dots, -1, 0, 1, \dots\}$, which represents that there are currently $|n|$ consumers (resp. providers) in the system if $n < 0$ (resp. $n > 0$), and no consumers or providers when $n = 0$. For a specific state n, we denote the signal that the platform broadcasts to user class u under information Regime I as $s_I^u(n)$. We denote the set of such signals as $S_I^u = \cup_n s_I^u(n)$.

The users' instantaneous equilibrium arrival rate depends on the signal $s_I^u(n)$, and we denote it by $\lambda_I^u(s_I^u(n))$. Accordingly, the long-run average arrival rate of users (consumers or providers) is obtained as the weighted average of the instantaneous arrival rates $\lambda_I^u(s_I^u(n))$, where the weights are the steady-state probabilities $\pi_I(n), n \in \mathbb{N}$; in turn, these probabilities depend on the information Regime I through the instantaneous arrival rates $\lambda_I^u(s_I^u(n))$ that I induces. Since any user who joins will be matched eventually, the long-run average arrival rate is equal to the effective match rate. Therefore, we can write the effective match rate under information Regime I as:

$$M_I = \sum_{n \in \mathbb{N}} \pi_I(n) \lambda_I^u(s_I^u(n)), \quad I \in \{\mathbb{O}, \mathbb{C}, \mathbb{P}\}, \tag{2.3}$$

where u may be either c or p. In Sections 2.4.1-2.4.2, we derive the effective match rates for all information regimes. Trivially, users can, in equilibrium, opt not to join given any signal, leading to an effective match rate of zero. Unless directly implied by the information regime, we disregard this trivial equilibrium joining strategy in the remainder of the chapter. If the system can accommodate a non-zero match rate, we assume that the users' equilibrium joining strategy will lead to this non-zero match rate.

2.4.1 Occupancy Information (Regime $\mathbb{O}$)

Under Regime $\mathbb{O}$, the platform provides the current system state to the arriving users; i.e., $s_{\mathbb{O}}^c(n) = s_{\mathbb{O}}^p(n) = n$, leading to $S_{\mathbb{O}}^c = S_{\mathbb{O}}^p = \mathbb{Z}$. A positive (resp. negative) signal indicates that providers (resp. consumers) are currently waiting to be matched. An alternative that results in identical joining behavior is to signal to providers (resp. consumers) the number of providers (resp. consumers) when there are no consumers (resp. providers) and to signal "no delay" when there are consumers (resp. providers). This roughly mimics the queue position information that ride-hailing platforms provide to drivers in airports (Paul, 2015), and also roughly mimics the expected delay information that those platforms provide to riders.

Under Regime $\mathbb{O}$, the expected delay of an arriving user from class u is entirely determined

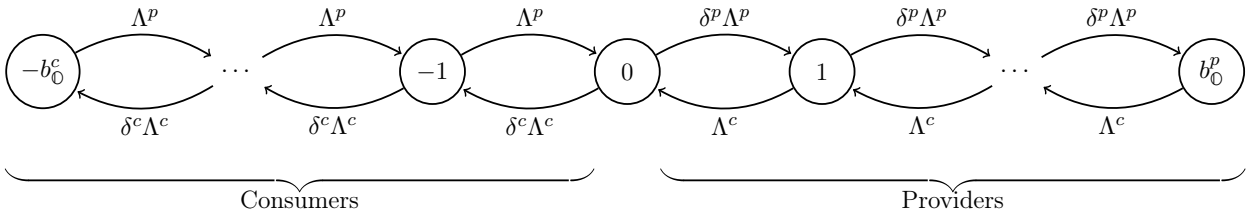


Figure 2.1: Regime $\mathbb{O}$ CTMC

by the signal s the user observes; the expected delay does not depend on the equilibrium joining probability $J_{\mathbb{O}}(\theta^u|s)$ of other users of the same class receiving signal s. Accordingly, each arriving user either balks or joins with probability one (i.e., there is no mixed strategy equilibrium). More explicitly, each arriving consumer who observes $s^c_{\mathbb{O}}(n) = n > 0$ faces a zero expected delay, regardless of the joining probability of other consumers in state n; therefore, they will join with probability one regardless of their type (i.e, whether their $\theta^c = t^c$ or $\theta^c = 1$), which leads to an instantaneous joining rate $\lambda^c_{\mathbb{O}}(s^c_{\mathbb{O}}(n)) = \Lambda^c$ for $n > 0$. Following a similar argument for providers, we have $\lambda^p_{\mathbb{O}}(s^p_{\mathbb{O}}(n)) = \Lambda^p, n < 0$. However, users join differently when their signal indicates a delay. Specifically, an arriving consumer who observes $s^c_{\mathbb{O}}(n) = n \leq 0$ faces an expected delay of $\mathbb{E}[W^c_{\mathbb{O}}(n)] = (|n| + 1)/\Lambda^p$ (each consumer ahead of them takes an exponentially distributed amount of time with mean $1/\Lambda^p$ to be matched). If this user is impatient, she will not join; otherwise, she joins with probability one when $|n|$ is sufficiently small. Accordingly, $\lambda^c_{\mathbb{O}}(s^c_{\mathbb{O}}(n)) = \delta^c\Lambda^c$ when $|n|$ is sufficiently small, and $\lambda^c_{\mathbb{O}}(s^c_{\mathbb{O}}(n)) = 0$ otherwise. Formally, patient consumers join with probability one if and only if $U^c_{\mathbb{O}}(n) \geq 0$, which yields the condition:

$$0 \leq |n| \leq \frac{\Lambda^p\,(1 - t^c)}{t^c} - 1. \tag{2.4}$$

By an analogous argument, when $n \geq 0$, impatient providers balk and patient providers join with probability one if and only if:

$$0 \leq n \leq \frac{\Lambda^c\,(1 - t^p)}{t^p K^p} - 1. \tag{2.5}$$

Observe in Eqs. (2.4)-(2.5) that patient consumers and providers only join until a threshold system state on their side of the platform (i.e., when the signal indicates a delay). Therefore, the resulting CTMC will be bounded on both sides, as illustrated in Fig. 2.1 where $b^c_{\mathbb{O}}$ and $b^p_{\mathbb{O}}$ are the respective *bounding states* on the consumers' and providers' sides. From Eqs. (2.4)-(2.5), those bounding states follow:

$$\begin{aligned} b^c_{\mathbb{O}} &= \lfloor \Lambda^p\,(1/t^c - 1) \rfloor\,, \\ b^p_{\mathbb{O}} &= \lfloor (\Lambda^c/K^p)\,(1/t^p - 1) \rfloor\,. \end{aligned} \tag{2.6}$$

Note that when $b^c_{\mathbb{O}} = 0$ (resp., $b^p_{\mathbb{O}} = 0$), patient consumers (resp., providers) balk even at state 0, i.e., they balk given any signal that indicates a non-zero expected delay. We characterize the instantaneous arrival rates and the effective match rates under Regime $\mathbb{O}$ in Proposition 2.1, depending on whether each bounding state is zero or not.

Proposition 2.1. *The instantaneous arrival rates and effective match rates in Regime $\mathbb{O}$ follow:*

Case	Condition	$\lambda^c_{\mathbb{O}}(s^c_{\mathbb{O}}(n))$		$\lambda^p_{\mathbb{O}}(s^p_{\mathbb{O}}(n))$		$M_{\mathbb{O}}$
		$-b^c_{\mathbb{O}} < n < 0 \vee n = 0$	$n > 0$	$n < 0$	$n = 0 \vee 0 < n < b^p_{\mathbb{O}}$	
1	$b^c_{\mathbb{O}} > 0$ & $b^p_{\mathbb{O}} > 0$	$\delta^c\Lambda^c$	Λ^c	Λ^p	$\delta^p\Lambda^p$	Eq. (2.7)
2	$b^c_{\mathbb{O}} > 0$ & $b^p_{\mathbb{O}} = 0$	$\delta^c\Lambda^c$	0	Λ^p	0	Eq. (2.8)
3	$b^c_{\mathbb{O}} = 0$ & $b^p_{\mathbb{O}} > 0$	0	Λ^c	0	$\delta^p\Lambda^p$	Eq. (2.9)
4	$b^c_{\mathbb{O}} = 0$ & $b^p_{\mathbb{O}} = 0$	0	0	0	0	0

where Eqs. (2.7)-(2.9) follow:

$$M_{\mathbb{O}} = \Lambda^p \frac{\delta^c\left(\Lambda^c - \delta^p\Lambda^p\right)\left(1 - \left(\delta^c\Lambda^c/\Lambda^p\right)^{b^c_{\mathbb{O}}}\right) + \delta^p\left(\Lambda^p - \delta^c\Lambda^c\right)\left(1 - \left(\delta^p\Lambda^p/\Lambda^c\right)^{b^p_{\mathbb{O}}}\right)}{\delta^c\left(\Lambda^c - \delta^p\Lambda^p\right)\left(1 - \left(\delta^c\Lambda^c/\Lambda^p\right)^{b^c_{\mathbb{O}}}\right) + \left(\Lambda^p - \delta^c\Lambda^c\right)\left(1 - \left(\delta^p\Lambda^p/\Lambda^c\right)^{b^p_{\mathbb{O}}+1}\right)}, \tag{2.7}$$

$$M_{\mathbb{O}} = \delta^c\Lambda^c + \frac{\Lambda^p - \delta^c\Lambda^c}{1 - \left(\Lambda^p/\delta^c\Lambda^c\right)^{b^c_{\mathbb{O}}+1}}, \tag{2.8}$$

$$M_{\mathbb{O}} = \Lambda^c - \frac{\Lambda^c - \delta^p\Lambda^p}{1 - \left(\delta^p\Lambda^p/\Lambda^c\right)^{b^p_{\mathbb{O}}+1}}. \tag{2.9}$$

2.4.2 Asymmetric Information Regimes (Regimes $\mathbb{C}$ and $\mathbb{P}$)

In the asymmetric regimes, the platform shares the current system state with one user class and shares no information with the other. For brevity, we present our analysis for the regime where the current system state information is shared with providers (Regime $\mathbb{P}$); we present results for Regime $\mathbb{C}$ at the end of this section. Under Regime $\mathbb{P}$, $s^p_{\mathbb{P}}(n) = n$ and $s^c_{\mathbb{P}}(n) = \emptyset$ (i.e., no information), $\forall n \in \mathbb{N}$. This leads to signal sets $S^p_{\mathbb{P}} = \mathbb{Z}$ and $S^c_{\mathbb{P}} = \emptyset$.

Since consumers receive no information under Regime $\mathbb{P}$, their joining behavior is identical in all system states, i.e., $\lambda^c_{\mathbb{P}}(s^c_{\mathbb{P}}(n)) = \lambda^c_{\mathbb{P}}(\emptyset), \forall n \in \mathbb{N}$. However, the providers' joining behavior is state-dependent. Similar to Regime $\mathbb{O}$, when the state indicates no delay for providers, i.e., $s^p_{\mathbb{P}}(n) = n < 0$, an arriving provider joins with probability one (independent of whether she is patient or impatient); hence, $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n)) = \Lambda^p$ for $n < 0$. On the other hand, if an arriving provider receives a signal $s^p_{\mathbb{P}}(n) = n \geq 0$, her expected delay depends on the equilibrium arrival rate of consumers (which is $\lambda^c_{\mathbb{P}}(\emptyset)$), and we have $\mathbb{E}[W^p_{\mathbb{P}}(n)] = (n+1)/\lambda^c_{\mathbb{P}}(\emptyset)$ (each provider already in the system and ahead of the arriving provider takes an exponentially distributed amount of time with mean $1/\lambda^c_{\mathbb{P}}(\emptyset)$ to be matched). If this provider is patient, she joins with probability one if n is sufficiently small;

if she is impatient, she balks. Accordingly, $\lambda^p_{\mathbb{P}}(n) = \delta^p \Lambda^p$ if n is sufficiently small, and $\lambda^p_{\mathbb{P}}(n) = 0$, otherwise. Formally, patient providers join with probability one if and only if $U^p_{\mathbb{P}}(n) \geq 0$, which yields the condition:

$$0 \leq n \leq \frac{\lambda^c_{\mathbb{P}}(\emptyset)\,(1-t^p)}{K^p t^p} - 1. \tag{2.10}$$

Based on (2.10), the bounding state $b^p_{\mathbb{P}}$ up to which patient providers join with probability one is given by:

$$b^p_{\mathbb{P}} = \lfloor \lambda^c_{\mathbb{P}}(\emptyset)\,(1-t^p)/(K^p t^p) \rfloor\,. \tag{2.11}$$

The bounding state $b^p_{\mathbb{P}}$ depends on the equilibrium arrival rate $\lambda^c_{\mathbb{P}}(\emptyset)$ of consumers given that they receive no information. We now explain how $\lambda^c_{\mathbb{P}}(\emptyset)$ is obtained. Any equilibrium with $\lambda^c_{\mathbb{P}}(\emptyset) > 0$ involves consumers joining at state 0 (otherwise, consumers would never join resulting in a trivial equilibrium with zero effective match rate). For consumers observing no information, joining at state 0 entails a positive conditional expected delay and joining at other states entails a non-negative conditional expected delay. Therefore, the expected delay of an arriving consumer observing no information will be positive, i.e., $E[W^c_{\mathbb{P}}(\emptyset)] > 0$. As a result, impatient consumers balk as $U^c_{\mathbb{P}}(\emptyset) = R^c\,(1 - 1\,(1 + E[W^c_{\mathbb{P}}(\emptyset)])) < 0$. On the other hand, patient consumers either join with a positive probability or balk completely, depending on the impact of their equilibrium joining behavior on their expected delay. Fig. 2.2 shows the resulting CTMC for Regime $\mathbb{P}$. The expected delay faced by an arriving consumer is given by:

$$E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset)] = \sum_{i=-\infty}^{0} \frac{|i|+1}{\Lambda^p} \pi_{\mathbb{P}}(i), \tag{2.12}$$

where the stationary probabilities $\pi_{\mathbb{P}}(i)$ can be obtained in terms of $b^p_{\mathbb{P}}$ and $\lambda^c_{\mathbb{P}}(\emptyset)$, which need to be found such that they are consistent with each other. For this, we examine different cases depending on the consumers' joining probability ($J_{\mathbb{P}}(t^c|\emptyset) = 0$, $0 < J_{\mathbb{P}}(t^c|\emptyset) < 1$, or $J_{\mathbb{P}}(t^c|\emptyset) = 1$) and the bounding state for providers ($b^p_{\mathbb{P}} = 0$ or $b^p_{\mathbb{P}} > 0$). For each case, we characterize the effective match rate in Proposition 2.2 using Eq. (2.3), which simply yields $M_{\mathbb{P}} = \lambda^c_{\mathbb{P}}(\emptyset)$ since the signal is the same in all states. For ease of exposition, we define $H^c = t^c/(1-t^c)$ and $H^p = K^p t^p/(1-t^p)$.

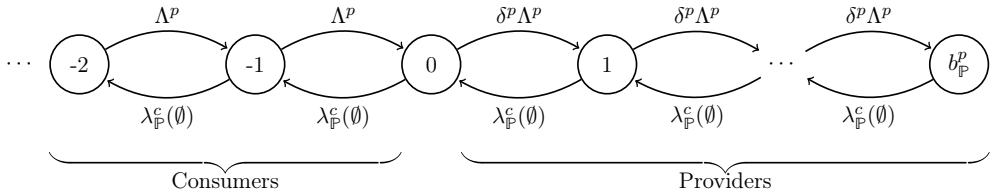


Figure 2.2: Regime $\mathbb{P}$ CTMC

Proposition 2.2. *Define the following cases: (1) $J_{\mathbb{P}}(t^c|\emptyset) = 1$, $b^p_{\mathbb{P}} > 0$, (2) $0 < J_{\mathbb{P}}(t^c|\emptyset) < 1$, $b^p_{\mathbb{P}} > 0$, (3) $J_{\mathbb{P}}(t^c|\emptyset) = 1$, $b^p_{\mathbb{P}} = 0$, (4) $0 < J_{\mathbb{P}}(t^c|\emptyset) < 1$, $b^p_{\mathbb{P}} = 0$, and (5) $J_{\mathbb{P}}(t^c|\emptyset) = 0$. The effective match rate under Regime $\mathbb{P}$ follows:*

Case	Conditions	$M_{\mathbb{P}}$
1	$\delta^c\Lambda^c < \Lambda^p \wedge H^c \leq \frac{(\Lambda^p-\delta^c\Lambda^c)\left(\delta^c\Lambda^c(1-\delta^p)-\delta^p(\Lambda^p-\delta^c\Lambda^c)(\delta^p\Lambda^p/\delta^c\Lambda^c)^{\lfloor\delta^c\Lambda^c/H^p\rfloor}\right)}{\delta^c\Lambda^c-\delta^p\Lambda^p} \wedge H^p \leq \delta^c\Lambda^c$	$\delta^c\Lambda^c$
2	$\left(\delta^c\Lambda^c \geq \Lambda^p \vee H^c \leq \frac{(\Lambda^p-\delta^c\Lambda^c)\left(\delta^c\Lambda^c(1-\delta^p)-\delta^p(\Lambda^p-\delta^c\Lambda^c)(\delta^p\Lambda^p/\delta^c\Lambda^c)^{\lfloor\delta^c\Lambda^c/H^p\rfloor}\right)}{\delta^c\Lambda^c-\delta^p\Lambda^p}\right) \wedge H^p \leq l_p$	m that solves (2.14)
3	$\delta^c\Lambda^c < \Lambda^p \wedge H^c \leq \Lambda^p - \delta^c\Lambda^c \wedge H^p > \delta^c\Lambda^c$	$\delta^c\Lambda^c$
4	$\left(\delta^c\Lambda^c \geq \Lambda^p \vee H^c > \Lambda^p - \delta^c\Lambda^c\right) \wedge H^c \leq \Lambda^p \wedge H^p + H^c > \Lambda^p$	$\Lambda^p - H^c$
5	Otherwise	0

where l_p in the condition of case 2 and Eq. (2.14) follow:

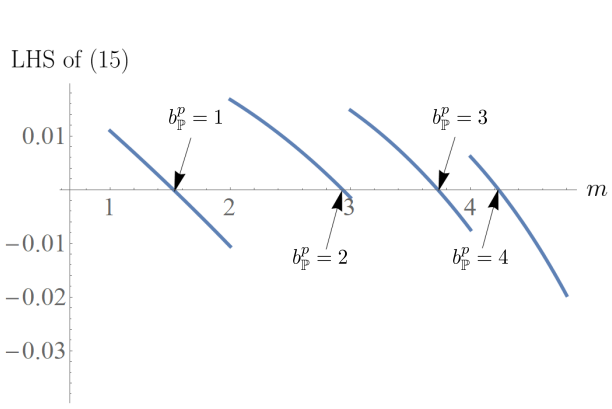
$$l_p = \frac{\Lambda^p(1-2\delta^p) - H^c + \sqrt{H^{c2} - 2\Lambda^p(1-2\delta^p)H^c + \Lambda^{p2}}}{2(1-\delta^p)}, \tag{2.13}$$

$$1 - \frac{H^c(\delta^p\Lambda^p - m)}{(\Lambda^p - m)\left(m(1-\delta^p) - \delta^p(\Lambda^p - m)\left(\delta^p\Lambda^p/m\right)^{\lfloor m/H^p\rfloor}\right)} = 0. \tag{2.14}$$

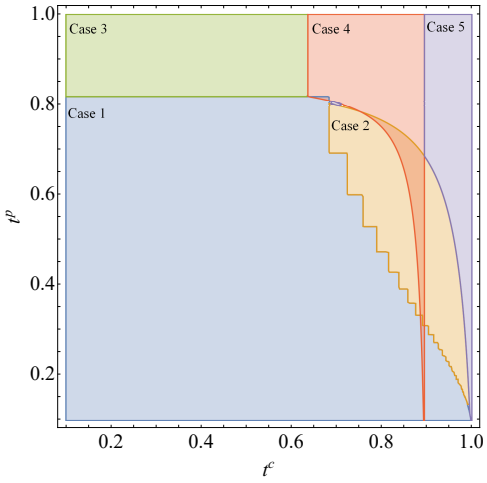
Two cases of overlaps occur among the cases of Proposition 2.2. First, the match rate associated with Case 2, which is the solution to Eq. (2.14), is not necessarily unique; i.e., Case 2 may yield multiple equilibria (multiple combinations of $b^p_{\mathbb{P}}$ and $\lambda^c_{\mathbb{P}}(\emptyset)$ that are consistent with each other). To illustrate, we plot the left hand side of Eq. (2.14) against m for a particular parameter setting in Fig. 2.3a, in which m crosses zero four times at $m \in \{1.53, 2.92, 3.75, 4.23\}$ corresponding to providers' bounding states $b^p_{\mathbb{P}} \in \{1, 2, 3, 4\}$. Second, the cases of Proposition 2.2 are not mutually exclusive, i.e., there are parameter settings under which more than one case holds, which again leads to multiple equilibria. We illustrate this in Fig. 2.3b, which highlights the regions corresponding to the different cases as a function of t^c and t^p. When there are multiple equilibria, we choose the equilibrium that produces a higher match rate. In Proposition 2.3, we list the possible overlaps between the cases of Proposition 2.2, and identify the case that results in a higher match rate.

Proposition 2.3. *There are only two possible overlaps between the cases of Proposition 2.2: (i) The conditions for cases 1 and 4 could hold together; if so, the effective match rate under Case 1 is higher. (ii) The conditions for cases 2 and 4 could hold together; if so, the effective match rate under Case 2 is higher.*

By interchanging the roles of consumers and providers in Proposition 2.2 and scaling time



(a) $\{\Lambda^c, \Lambda^p, K^p, t^c, t^p, \delta^c, \delta^p\} = \{8, 9, 1.5, 0.97, 0.4, 0.6, 0.68\}$



(b) $\{\Lambda^c, \Lambda^p, K^p, \delta^c, \delta^p\} = \{9, 8.5, 1.5, 0.75, 0.9\}$

Figure 2.3: Cases of Multiple Equilibria in Regime $\mathbb{P}$

appropriately (we describe this process in Appendix B.1), we can derive the match rates under Regime $\mathbb{C}$ as in Proposition 2.4.

Proposition 2.4. *Define the following cases: (1)* $J_{\mathbb{C}}(t^p|\emptyset) = 1$, $b^c_{\mathbb{C}} > 0$, *(2)* $0 < J_{\mathbb{C}}(t^p|\emptyset) < 1$, $b^c_{\mathbb{C}} > 0$, *(3)* $J_{\mathbb{C}}(t^p|\emptyset) = 1$, $b^c_{\mathbb{C}} = 0$, *(4)* $0 < J_{\mathbb{C}}(t^p|\emptyset) < 1$, $b^c_{\mathbb{C}} = 0$, *and (5)* $J_{\mathbb{C}}(t^p|\emptyset) = 0$. *The effective match rate under Regime* $\mathbb{C}$ *follows:*

Case	Conditions	$M_{\mathbb{C}}$
1	$\delta^p\Lambda^p < \Lambda^c \wedge H^p \leq \dfrac{\delta^c(\Lambda^c - \delta^p\Lambda^p)\left(\frac{\delta^c\Lambda^c}{\delta^p\Lambda^p}\right)^{\lfloor \delta^p\Lambda^p/H^c \rfloor} - \delta^p\Lambda^p(1-\delta^c)}{\delta^c\Lambda^c/(\delta^p\Lambda^p) - 1} \wedge H^c \leq \delta^p\Lambda^p$	$\delta^p\Lambda^p$
2	$\left(\delta^p\Lambda^p \geq \Lambda^c \vee H^p > \dfrac{\delta^c(\Lambda^c - \delta^p\Lambda^p)\left(\frac{\delta^c\Lambda^c}{\delta^p\Lambda^p}\right)^{\lfloor \delta^p\Lambda^p/H^c \rfloor} - \delta^p\Lambda^p(1-\delta^c)}{\delta^c\Lambda^c/(\delta^p\Lambda^p) - 1}\right) \wedge H^c \leq l_c$	m that solves (2.16)
3	$\delta^p\Lambda^p < \Lambda^c \wedge H^p \leq \Lambda^c - \delta^p\Lambda^p \wedge H^c > \delta^p\Lambda^p$	$\delta^p\Lambda^p$
4	$(\delta^p\Lambda^p < \Lambda^c \vee H^p > \Lambda^c - \delta^p\Lambda^p) \wedge H^p \leq \Lambda^c \wedge H^c + H^p > \Lambda^c$	$\Lambda^c - H^p$
5	Otherwise	0

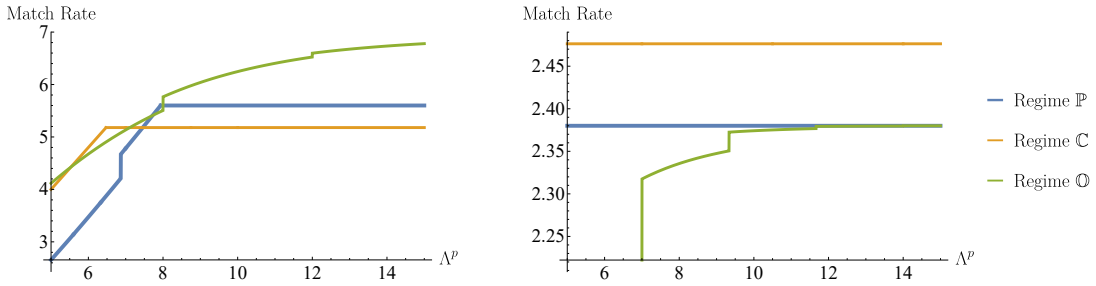
where l_c *and Eq. (2.16) follow:*

$$l_c = \frac{\Lambda^c(1-2\delta^c) - H^p + \sqrt{H^{p2} - 2\Lambda^c(1-2\delta^c)H^p + \Lambda^{c2}}}{2(1-\delta^c)}, \tag{2.15}$$

$$1 - \frac{H^p(\delta^c\Lambda^c - m)}{(\Lambda^c - m)\left(m(1-\delta^c) - \delta^c(\Lambda^c - m)\left(\delta^c\Lambda^c/m\right)^{\lfloor m/H^c \rfloor}\right)} = 0. \tag{2.16}$$

2.5 Platform's Optimal Information Regime

Equipped with the match rate expressions derived in Section 2.4, we now compare the match rates under the three regimes in order to determine which regime maximizes the platform's effective match rate. A general characterization of when one information regime outperforms the others is complicated due to the conditions governing the case-dependent match rates (given in Propositions 2.1, 2.2, and 2.4) and the intractability of Eqs. (2.14) and (2.16). Fig. 2.4a illustrates this complication: for fixed values of all other parameters, the optimal regime switches several times from one regime to another as Λ^p increases. (We observe a similar phenomenon with respect to the other parameters.) Furthermore, unlike Fig. 2.4a in which Regime $\mathbb{O}$ outperforms the other regimes for most Λ^p values, Regime $\mathbb{O}$ is sub-optimal for *all* Λ^p values in the example shown in Fig. 2.4b. Although these difficulties preclude the possibility of obtaining necessary *and* sufficient conditions under which each regime is optimal, we present sufficient conditions under which each regime is optimal in Proposition 2.5.



(a) $\{t^c, t^p, \delta^c, \delta^p\} = \{0.8, 0.7, 0.8, 0.8\}$ (b) $\{t^c, t^p, \delta^c, \delta^p\} = \{0.7, 0.88, 0.34, 0.67\}$

Figure 2.4: Complicated structure of the optimal information regime; illustration for $\Lambda^c = 7$, $K^p = 1$

Proposition 2.5. *(i) Under the conditions for Case 1 of Proposition 2.2 (resp., Proposition 2.4), if* $\delta^c\Lambda^c > \delta^p\Lambda^p$ *(resp.,* $\delta^p\Lambda^p > \delta^c\Lambda^c$*) and* (2.17) *(resp.* (2.18)*) holds, then Regime* $\mathbb{P}$ *(resp. Regime* $\mathbb{C}$*) is optimal. (ii) Regime* $\mathbb{O}$ *is optimal if neither of* (2.17) *and* (2.18) *holds.*

$$\delta^c \left(\Lambda^c\delta^c/\Lambda^p\right)^{b^c_{\mathbb{O}}} > \frac{\delta^p\Lambda^p(1-\delta^c)}{\Lambda^c - \delta^p\Lambda^p}\left(1 - \left(\Lambda^p\delta^p/\Lambda^c\right)^{b^p_{\mathbb{O}}}\right) \tag{2.17}$$

$$\delta^p \left(\Lambda^p\delta^p/\Lambda^c\right)^{b^p_{\mathbb{O}}} > \frac{\delta^c\Lambda^c(1-\delta^p)}{\Lambda^p - \delta^c\Lambda^c}\left(1 - \left(\Lambda^c\delta^c/\Lambda^p\right)^{b^c_{\mathbb{O}}}\right) \tag{2.18}$$

We now explain the intuition behind the sufficient conditions for the optimality of Regime $\mathbb{P}$ in Proposition 2.5; the intuition behind the conditions for Regime $\mathbb{C}$ and $\mathbb{O}$ follow similarly. When the Case 1 conditions of Proposition 2.2 hold, the match rate of Regime $\mathbb{P}$ is $\delta^c\Lambda^c$. When $\delta^c\Lambda^c > \delta^p\Lambda^p$, Regime $\mathbb{P}$ outperforms the best possible match rate of Regime $\mathbb{C}$ ($\delta^p\Lambda^p$). So it remains to compare Regime $\mathbb{P}$ to Regime $\mathbb{O}$. In doing so, observe that switching from Regime $\mathbb{P}$ to Regime $\mathbb{O}$ results in two major changes that impact the match rate: (i) Impatient consumers will join when they observe

zero delay, thereby inducing providers to be willing to join when the queue is longer; and (ii) There will be a maximum queue length on the consumers' side beyond which patient consumers will not join (i.e., the consumers' bounding state becomes finite). While the first change favors Regime $\mathbb{O}$, the second change favors Regime $\mathbb{P}$. When the first effect is insignificant (i.e., when there are not many impatient consumers), the net effect of both changes will be in favor of Regime $\mathbb{P}$. Accordingly, there is a threshold on δ^c above which Regime $\mathbb{P}$ is preferable to Regime $\mathbb{O}$. Ineq. (2.17) reflects this intuition: the left hand side of Ineq. (2.17) is increasing in δ^c and the right hand side of Ineq. (2.17) is decreasing in δ^c.

The sufficient conditions in Proposition 2.5 show that providing more information can, in fact, hurt the platform. While these conditions provide a partial characterization of when (and why) information can hurt, we further investigate the system under limiting settings to better understand when the platform prefers one of the asymmetric information regimes, i.e., when information can hurt. In Section 2.5.1, we examine a situation where the patience profile discrepancy between the two user classes (which is based on t^c and t^p) is taken to the limit. Subsequently, in Section 2.5.2, we study a situation where the market size on one side (which is based on Λ^c and Λ^p) is taken to the limit.

2.5.1 Patience Profile Discrepancy

In this section, we demonstrate that the discrepancy in patience profiles between providers and consumers determines whether disclosing more information (Regime $\mathbb{O}$) hurts the platform. Specifically, in Proposition 2.6 we show that when there is a large discrepancy between the patience profiles of providers and consumers, it is optimal to hide information from one of them.

Proposition 2.6. *(i) When $t^p \in (\frac{\Lambda^c}{K^p+\Lambda^c}, 1)$, $t^c \to 0$ and $\delta^c\Lambda^c < \delta^p\Lambda^p$, Regime $\mathbb{C}$ is optimal for the platform. (ii) When $t^c \in (\frac{\Lambda^p}{1+\Lambda^p}, 1)$, $t^p \to 0$ and $\delta^p\Lambda^p < \delta^c\Lambda^c$, Regime $\mathbb{P}$ is optimal for the platform.*

We now explain the intuition behind Proposition 2.6(i); the intuition behind Proposition 2.6(ii) is analogous. First, note that since t^p is large, providers, when given information, do not join at state 0. Accordingly, impatient consumers (whether they are given information or not) never join the system, leading to a maximum possible match rate of $\delta^c\Lambda^c$ under Regimes $\mathbb{O}$ and $\mathbb{P}$. On the other hand, under Regime $\mathbb{C}$, providers can be induced to join at a higher rate than $\delta^c\Lambda^c$ regardless of state (if there are enough patient providers to allow this), because patient consumers, who have a very low delay sensitivity ($t^c \to 0$), join up to a very high bounding state, driving the expected delay of providers to zero.

While Proposition 2.6 presents results for limiting scenarios (i.e., taking t^c or t^p to the limit), we check whether the insight that a high patience profile discrepancy leads to an asymmetric regime being optimal holds more generally. To do this, we perform a large set of numerical experiments: We compute the match rates under Regimes $\mathbb{O}$, $\mathbb{C}$, and $\mathbb{P}$ for the following parameters: $\Lambda^c \in \{1.1, 5, 10.1, 15\}$, $\Lambda^p \in \{1, 5.1, 10, 15.1\}$, $K^p = 1$, $t^c \in \{0.15, 0.35, 0.55, 0.75, 0.95\}$, $t^p \in$

$\{0.15, 0.35, 0.55, 0.75, 0.95\}$, $\delta^c \in \{0.15, 0.35, 0.55, 0.75, 0.95\}$ and $\delta^p \in \{0.15, 0.35, 0.55, 0.75, 0.95\}$, for a total of 10,000 experiments. We define more information as hurting the system if the match rate under Regime $\mathbb{C}$ or $\mathbb{P}$ is at least τ% higher than the match rate under Regime $\mathbb{O}$. A positive value of τ allows us to draw meaningful insights on Regime $\mathbb{O}$'s possible sub-optimality, by isolating only those cases where Regime $\mathbb{O}$ is *far from optimal* (i.e., it is not near-optimal). We use a classification tree to understand under what settings more information hurts; we denote the classes as $\mathcal{H}$ (Regime $\mathbb{O}$ hurts) and $\mathcal{N}$ (Regime $\mathbb{O}$ does not hurt as is it near optimal). Setting τ poses a trade-off: a higher value of τ causes fewer parameter settings to be classified as $\mathcal{H}$, while those settings that are labelled as $\mathcal{H}$ correspond to Regime $\mathbb{O}$ hurting more strongly. We try $\tau \in \{5, 8, 10\}$, and we find that $\tau = 8$ yields a reasonable accuracy and an interpretable classification tree. Fig. 2.5 presents the resulting classification tree, where the nodes labeled 1 to 4 highlight the parameter settings that are classified as $\mathcal{H}$. For nodes 1 and 2, Regime $\mathbb{C}$ is preferred by the platform for 70% and 86% of experiments, respectively, while for nodes 3 and 4 Regime $\mathbb{P}$ is preferred by the platform for 87% and 96% of the experiments, respectively. In the figure, class $\mathcal{H}$ nodes correspond to the following parameter settings:

- Nodes 1, 2: $\delta^p \geq 0.55$, $t^p = 0.95$, $\delta^c \leq 0.35$, $\Lambda^c \geq 10.1$, and $t^c \leq 0.35, \Lambda^p \geq 5.1$ (Node 1) or $0.55 \leq t^c \leq 0.75$, $\Lambda^p = 15.1$ (Node 2): These can be interpreted as relaxed versions of the conditions presented in Proposition 2.6(i), where the additional conditions on Λ^c and Λ^p ensure that the effects driven by the discrepancy are strong enough.

- Nodes 3, 4: $\delta^c \geq 0.35$, $t^c = 0.95$, $\delta^p = 0.15$, $\Lambda^p \geq 10$, and $t^p \leq 0.35$, $\Lambda^c \geq 5$ (Node 3) or $0.55 \leq t^p \leq 0.75$, $\Lambda^c = 15$ (Node 4): These can be interpreted as relaxed versions of the conditions presented in Proposition 2.6(ii), where the additional conditions on Λ^c and Λ^p ensure that the effects driven by the discrepancy are strong enough.

In summary, the results of our classification tree generalize the insight we obtained in Proposition 2.6: a high patience profile discrepancy results in more information hurting the platform. Accordingly, for a platform serving user classes with very different delay sensitivities, it is (perhaps counter-intuitively) optimal to hide occupancy information from the user class with highly delay sensitive users.

2.5.2 Market Size Imbalance

In this section, we examine how the match rates compare under a limiting scenario where the arrival rate of one user class is unbounded. For ease of exposition, we consider the case of $\Lambda^c \to \infty$; results for the other case ($\Lambda^p \to \infty$) can be derived analogously. We now discuss the impact of $\Lambda^c \to \infty$ on each regime's performance. Under Regime $\mathbb{C}$, patient providers do not receive any delay information upon arrival and they join at rate $\delta^p \Lambda^p$ when $\Lambda^c \to \infty$ (regardless of their delay sensitivity) as they expect to be matched almost instantly; this leads to a match rate of $\delta^p \Lambda^p$. Under Regime $\mathbb{P}$ when $\Lambda^c \to \infty$, consumers employ a mixed strategy to avoid crowding; this leads to a match rate

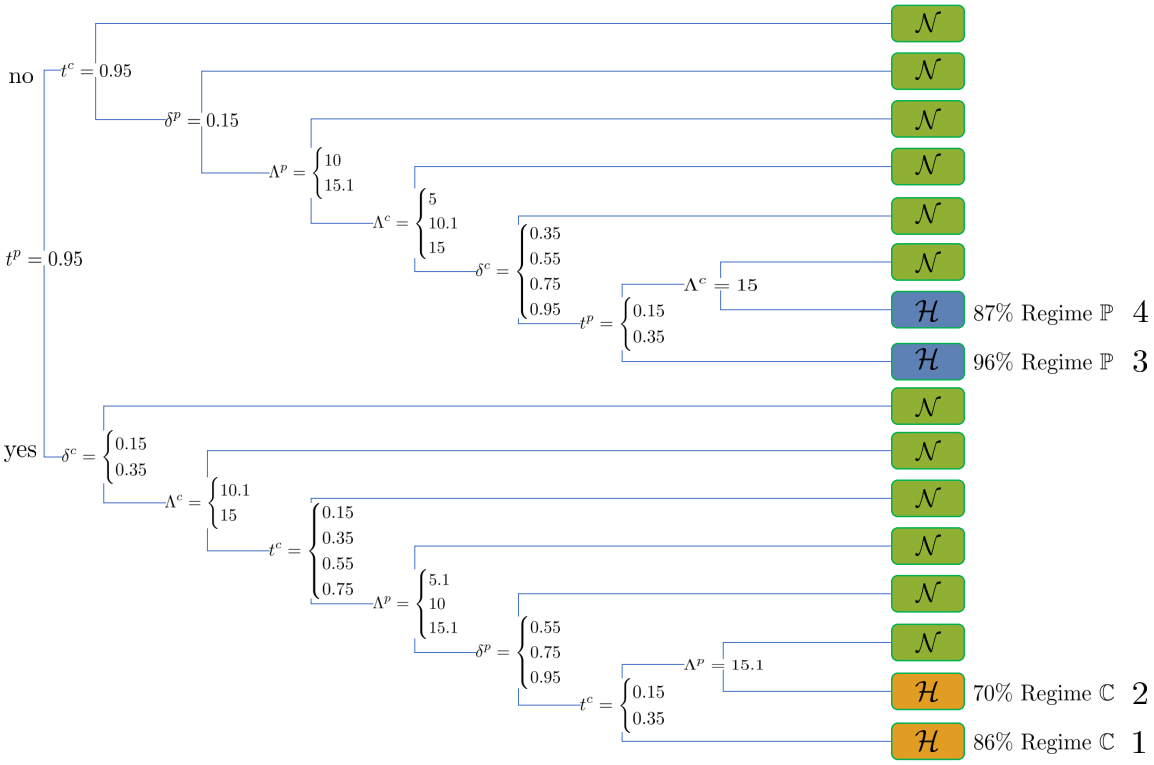


Figure 2.5: Classification tree for when occupancy information hurts the platform.

between 0 and Λ^p depending on the delay sensitivity of the patient consumers. Under Regime $\mathbb{O}$ when $\Lambda^c \to \infty$, the joining behavior of patient consumers at state 0 determines the match rate: If they are patient enough, the CTMC in Fig. 2.1 has its mass concentrated at a non-zero bounding state $-b^c_{\mathbb{O}}$ leading to a match rate of Λ^p; otherwise, the mass is concentrated at state 0 leading to a match rate of $\delta^p\Lambda^p$. We formalize this discussion in Lemma 2.7.

Lemma 2.7. *(i)* $\lim_{\Lambda^c\to\infty} M_{\mathbb{C}} = \delta^p\Lambda^p$. *(ii)* $\delta^p\Lambda^p < \lim_{\Lambda^c\to\infty} M_{\mathbb{P}} < \Lambda^p$ *if* $t^c < T_{\mathbb{P}}$ *(given in (2.19)) and* $0 < \lim_{\Lambda^c\to\infty} M_{\mathbb{P}} < \delta^p\Lambda^p$, *otherwise. (iii)* $\lim_{\Lambda^c\to\infty} M_{\mathbb{O}} = \Lambda^p$ *if* $t^c < T_{\mathbb{O}}$ *(given in (2.20)) and* $\lim_{\Lambda^c\to\infty} M_{\mathbb{O}} = \delta^p\Lambda^p$, *otherwise.*

$$T_{\mathbb{P}} = \left(1 + \frac{1}{\Lambda^p(1-\delta^p)\left((1-\delta^p)\left\lfloor \frac{\delta^p\Lambda^p(1-t^p)}{K^p t^p}\right\rfloor + 1\right)}\right)^{-1}, \tag{2.19}$$

$$T_{\mathbb{O}} = \frac{\Lambda^p}{1+\Lambda^p}. \tag{2.20}$$

Using the characterization in Lemma 2.7, we compare the match rates among the three information regimes in Proposition 2.8.

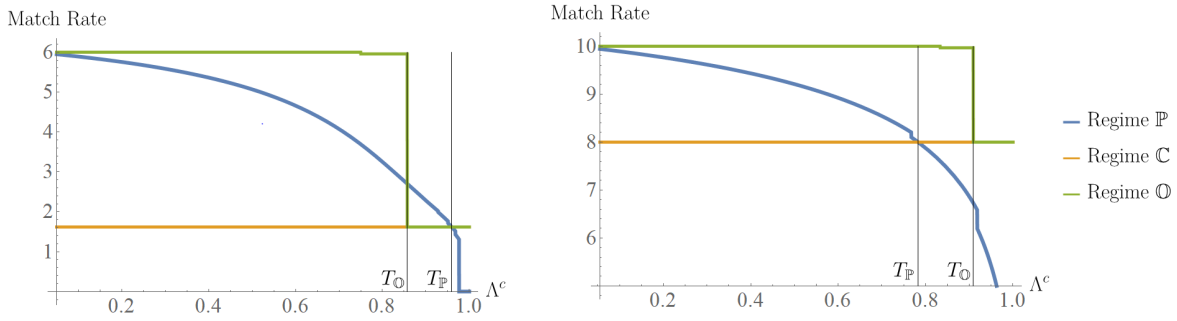
Proposition 2.8. *When* $\Lambda^c \to \infty$, *the effective match rates under the three information regimes are ordered as follows:*

Case	Conditions	Ordering
1	$t^c < \min\{T_{\mathbb{P}}, T_{\mathbb{O}}\}$	$M_{\mathbb{O}} > M_{\mathbb{P}} > M_{\mathbb{C}}$
2	$T_{\mathbb{P}} \le t^c < T_{\mathbb{O}}$	$M_{\mathbb{O}} > M_{\mathbb{C}} > M_{\mathbb{P}}$
3	$T_{\mathbb{O}} \le t^c < T_{\mathbb{P}}$	$M_{\mathbb{P}} > M_{\mathbb{O}} = M_{\mathbb{C}}$
4	$t^c \ge \max\{T_{\mathbb{P}}, T_{\mathbb{O}}\}$	$M_{\mathbb{O}} = M_{\mathbb{C}} > M_{\mathbb{P}}$

We illustrate the findings of Proposition 2.8 in Fig. 2.6, where we plot the match rates under the three information regimes against the consumer delay sensitivity t^c when $\Lambda^c = 1000$. Fig 2.6a displays a scenario where $T_{\mathbb{P}} > T_{\mathbb{O}}$ (Case 2 does not apply for any t^c). When $t^c < T_{\mathbb{O}}$, consumers are patient enough to join at state 0 under Regime $\mathbb{O}$, and therefore, providers almost always arrive to a system with zero delay. This in turn leads to Regime $\mathbb{O}$ achieving the highest possible match rate (i.e., Λ^p). When $t^c = T_{\mathbb{O}}$, $M_{\mathbb{O}}$ drops sharply to $\delta^p\Lambda^p$ because consumers now balk at state 0, inducing impatient providers to always balk; this leads to Regime $\mathbb{P}$ being optimal. As t^c continues to increase, Regime $\mathbb{P}$ continues to be optimal (but with decreasing match rates, as patient consumers join with lower probability) until $t^c = T_{\mathbb{P}}$, at which point $M_{\mathbb{P}} = M_{\mathbb{O}}$. A further increase in t^c causes $M_{\mathbb{P}}$ to

continue to drop, while $M_{\mathbb{O}}$ remains stable at $\delta^p \Lambda^p$. Therefore, for the setting shown in Fig 2.6a, Regime $\mathbb{O}$ yields the (weakly) best match rate, except when $T_{\mathbb{O}} \leq t^c < T_{\mathbb{P}}$.

On the other hand, Fig 2.6b displays a scenario where $T_{\mathbb{P}} < T_{\mathbb{O}}$ (Case 3 does not apply for any t^c). Here, Regime $\mathbb{O}$ yields the (weakly) best match rate for all values of t^c. This is because the discontinuous drop in the match rate of Regime $\mathbb{O}$ occurs for a value of t^c large enough that the match rate under Regime $\mathbb{P}$ is already below $\delta^p \Lambda^p$.



(a) $\Lambda^p = 6,\ K^p = 0.41,\ \delta^c = 0.6,\ \delta^p = 0.27, t^p = 0.38$

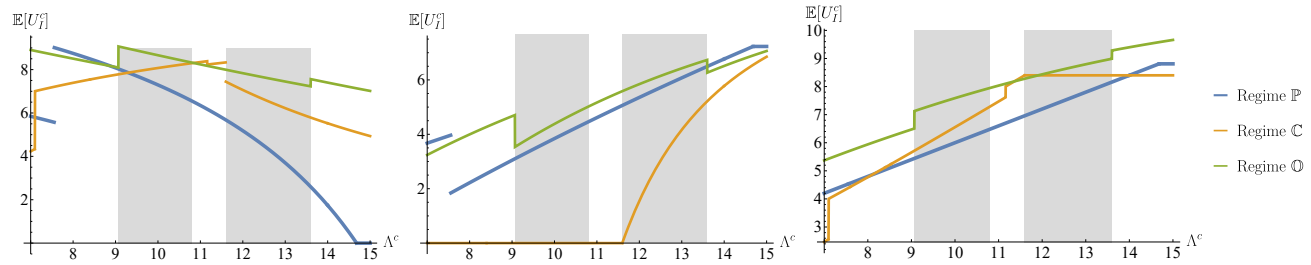
(b) $\Lambda^p = 10,\ K^p = 0.41,\ \delta^c = 0.6,\ \delta^p = 0.8, t^p = 0.8$

Figure 2.6: Match Rate plots for $\Lambda^c = 1000$

In summary, when consumers are plentiful compared to providers, the platform always enjoys a higher match rate when it informs providers of the abundance of consumers. However, whether the platform benefits from sharing or hiding information from consumers depends on the patient consumers' delay sensitivity. When this delay sensitivity is very low, sharing information can take advantage of the consumers' willingness to join, while hiding information from them forgoes this opportunity. Similarly, when the delay sensitivity is very high, sharing information with consumers induces them to join when their delay is relatively low, while hiding information from them dissuades them from joining with high probability; therefore, the platform benefits from sharing information with consumers. These effects lead to the insight that when consumers are plentiful, sharing information with them is optimal when consumers are either very patient or very impatient. On the other hand, when consumers are neither very patient nor very impatient, hiding information from them may turn out to be optimal depending on the parameter setting.

2.5.3 Comparison with Users' Preferences

We now study the effect of the platform's regime choice on the users by examining the users' welfare under the three information regimes. First, we derive expressions for the users' welfare under each regime. Then, we numerically study whether and when the platform's choice of optimal regime also maximizes the welfare of both user classes (we call this situation *full alignment*). The expected



(a) Consumers' Welfare (b) Providers' Welfare (c) Match Rate for the Platform

Figure 2.7: $\Lambda^p = 12, R^c = R^p = 100, K^p = 0.8, t^c = 0.8, t^p = 0.85, \delta^c = 0.6, \delta^p = 0.7$.

welfare of user class $u \in \{c, p\}$ under information Regime $I \in \{\mathbb{O}, \mathbb{C}, \mathbb{P}\}$ is given by

$$\mathbb{E}[U_I^u] = \sum_{s \in S_I^u} \Pr(s) \mathbb{E}[U_I^u(s)], \tag{2.21}$$

where $\Pr(s)$ is the probability that the user class u receives signal s under information Regime I. Since the expressions for users' welfare under the three information regimes are unwieldy, we present them in Appendix B.3. We use these expressions to characterize the platform's and users' possible alignment. We perform this analysis numerically, as we do not have closed-form expressions for the users' utilities under Case 2 of Regimes $\mathbb{C}$ and $\mathbb{P}$.

Fig. 2.7 presents a representative example of our numerical analysis for the three information regimes. For consumers and providers, we plot the expected welfare against Λ^c in Figs. 2.7a-2.7b, and for the platform, we plot the effective match rate against Λ^c in Fig. 2.7c. As seen in Figs. 2.7a-2.7b, a user class might receive the highest utility in a regime where information is hidden from one user class. For example, in the setting of Fig. 2.7a, consumers are best-off when information is hidden from providers when Λ^c is between approximately 10.8 and 11.6. It is even more interesting to observe that a user class may be best-off when information is hidden from their *own* class; for instance, this occurs for consumers when Λ^c is approximately between 7.5 and 9.1 in Fig. 2.7a. We can understand why this is the case by examining the CTMCs in Figs. 2.1 and 2.2. Changing from Regime $\mathbb{P}$ to $\mathbb{O}$ has three different effects: (1) The bounding state for providers is higher under Regime $\mathbb{O}$ than under Regime $\mathbb{P}$ ($b^p_{\mathbb{O}} > b^p_{\mathbb{P}}$). (2) The bounding state for consumers, $b^c_{\mathbb{O}}$, becomes finite (this bounding state is infinite under Regime $\mathbb{P}$). Both (1) and (2) reduce the expected delay (and hence increase the utility) of consumers. (3) The rate at which providers are cleared from the system increases, because consumers now join at a rate $\Lambda^c > \lambda^c_{\mathbb{P}}(\emptyset)$ when providers are in excess: this causes consumers to be less likely to arrive at a system with providers, and therefore, decreases their utility. When the third effect is stronger than the first two, consumers obtain higher expected utility under Regime $\mathbb{P}$ than under Regime $\mathbb{O}$. Our observation that users might prefer less information in a two-sided setting complements a similar effect noted for one-sided settings in Guo and Zipkin (2007).

Although a user class's welfare may be maximized under a regime where one user class (as

explained, possibly themselves) receives no information, either the other user class or the platform does not find the same regime optimal. Specifically, Fig. 2.7 illustrates our general observation that *full alignment*, when it occurs, only occurs for Regime $\mathbb{O}$ (the shaded regions in the figure correspond to parameters where the same regime yields maximal utility for users and maximal match rate for the platform). As a corollary, when the platform finds Regime $\mathbb{C}$ (resp., Regime $\mathbb{P}$) optimal, at least one user class favors Regime $\mathbb{P}$ or Regime $\mathbb{O}$ (resp., Regime $\mathbb{C}$ or Regime $\mathbb{O}$).

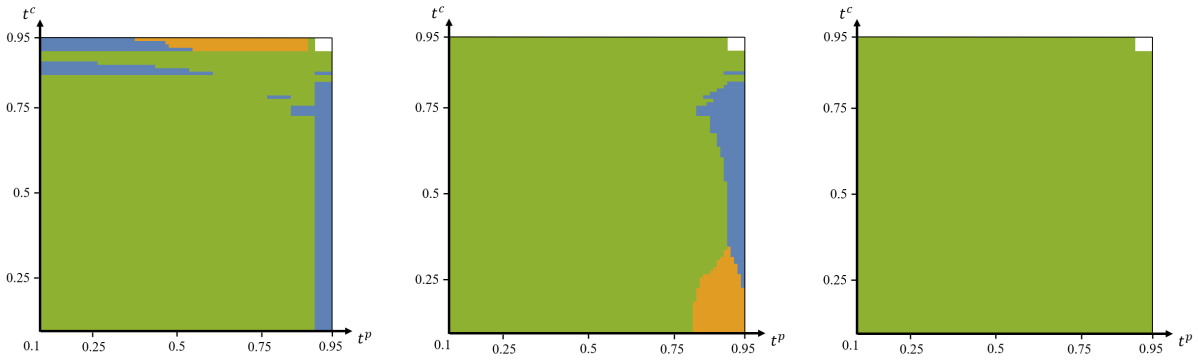
2.6 Extension to More Heterogeneous User Populations

In this section, we examine how the choice of the optimal information regime varies as the user populations become more heterogeneous with respect to their delay sensitivity. If one thinks of different values that users' delay sensitivities could take as the degree of heterogeneity in patience levels (similar to Guo and Zipkin, 2007), the distribution that models the most heterogeneous user population is a continuous distribution. We restrict our attention to uniform distributions, and we compare the optimal regime choice under our base model (a two-point uniform distribution) to that under a continuous uniform distribution with the same mean (to allow for a fair comparison). Formally, for user class $u \in \{c, p\}$ and $t^u \in [0, 1)$, the two distributions follow:

$$\begin{aligned} f_2^u(\theta^u) &= 1/2 \quad \text{for} \quad \theta^u \in \{t^u, 1\}, \\ f_{uni}^u(\theta^u) &= 1/(1-t^u) \quad \text{for} \quad \theta^u \in [t^u, 1]\,. \end{aligned} \tag{2.22}$$

We consider three different settings: (a) Two-point: both providers' and consumers' delay sensitivities follow uniform two-point distributions; (b) hybrid: consumers' delay sensitivities follow the continuous uniform distribution while providers' delay sensitivities follow the uniform two-point distribution (for conciseness, we skip the symmetric analogue of this setting); (c) continuous: both user classes' delay sensitivities follow continuous uniform distributions. Going from setting (a) to (c) represents an increase in the extent of heterogeneity in the users' delay sensitivities.

Fig. 2.8 depicts the optimal information regime for the three mentioned settings as t^c and t^p vary, for a representative setting of Λ^c, Λ^p and K^p. Note that both the two-point and hybrid settings exhibit a complicated optimal information regime structure, while in the continuous setting, Regime $\mathbb{O}$ outperforms the other regimes for all t^c and t^p values. We observe in Fig. 2.8 that as we switch from the two-point setting to the hybrid setting and then to the continuous setting, Regime $\mathbb{O}$ is optimal for a wider set of parameters (although there are specific combinations of t^c and t^p for which Regime $\mathbb{O}$ is optimal for the two-point setting, while Regime $\mathbb{C}$ or $\mathbb{P}$ is optimal for the hybrid setting). To verify whether these observations hold more generally, we conduct 3125 experiments with parameters $\Lambda^c \in \{1, 5, 10, 15, 20\}$, $\Lambda^p \in \{1.5, 5.5, 10.5, 15.5, 20.5\}$, $K^p \in \{0.2, 0.7, 1, 1.9, 6\}$, $t^c \in \{0.15, 0.35, 0.55.0.75, 0.95\}$ and $t^p \in \{0.15, 0.35, 0.55.0.75, 0.95\}$. Of these, we remove the 165 experiments in which all three regimes result in an effective match rate of 0, leaving us with 2960



(a) Two-point (b) Hybrid (c) Continuous

Figure 2.8: Optimal regimes; $\Lambda^c = 15, \Lambda^p = 10.5,\ K^p = 1.5$;
(Blue, Orange, Green) correspond to ($\mathbb{P}$, $\mathbb{C}$, $\mathbb{O}$). The uncolored top-right corners are regions where the match rate is zero in all three information regimes.

Table 2.1: Number of experiments under which each regime is optimal

	Two-point	**Hybrid**	**Continuous**
Regime $\mathbb{P}$	555	512	0
Regime $\mathbb{C}$	413	69	0
Regime $\mathbb{O}$	1992	2379	2960

experiments. Aligned with our observations in Fig. 2.8, we find that Regime $\mathbb{O}$ is optimal in more experiments as the patience levels become more heterogeneous (as with Fig. 2.8, there are, however, specific experiments for which Regime $\mathbb{O}$ is optimal in the two-point setting and either Regime $\mathbb{P}$ or $\mathbb{C}$ is optimal in the hybrid setting). Furthermore, we verify that Regime $\mathbb{O}$ is always optimal in the continuous setting. Table 2.1 summarizes our experimental results.

The main insight from these experiments is that as the users' heterogeneity (the number of distinct user types) increases, the platform tends to prefer sharing occupancy information. This complements the findings of Guo and Zipkin (2007) and Dobson and Pinker (2006) for one-sided settings. For example, Guo and Zipkin (2007) conclude that sharing occupancy information (or, in their terminology, "partial information") outperforms no information sharing when "the cost-scale distribution is spread out, so customers are heterogeneous." Similarly, Dobson and Pinker (2006) find that a firm benefits from sharing more detailed lead time information when customers' tolerances for waiting are sufficiently heterogeneous.

2.7 Conclusion

In this chapter, we study an on-demand platform's optimal delay information disclosure when the platform matches two classes of users (consumers and providers) with the objective of maximizing the effective match rate (as a proxy for revenue). We study three information regimes— *occupancy*, where both consumers and providers are informed of the current system occupancy, and two asymmetric information regimes, where either only consumers or only providers are informed of the current system occupancy, while the other is provided with no information.

Our base model considers users of each class as being either patient or impatient; we analytically compute and compare the platform's match rates under the three information regimes and provide sufficient conditions for each regime's optimality. We further find that it is optimal for platform to share no delay information with its users under certain limiting settings. Through numerical analysis, we show that user's welfare is not always aligned with platform's profitability and only aligned under occupancy information regime. We extend our base model by studying how the platform's information regime choice changes with higher levels of user heterogeneity (a larger number of distinct patience sensitivity levels). As users become more heterogeneous, the platform finds it optimal to disclose occupancy information to at least one of its user classes for a wider range of parameters.

We discuss a few interesting directions in which our models may be extended. First, it would be interesting to study the effect of users having non-linear delay cost functions. This analytically difficult extension would supplement the work of Guo and Zipkin (2007), who study the effect of customer delay cost functions on information disclosure in a one-sided system. Second, the bulk of our study is restricted to studying the platform's decision when user delay sensitivity follows a two-point distribution. Although we consider an extension to three-point and uniform distributions, it will be useful to find a general link between the distribution of user delay sensitivity and the firm's optimal policy; this problem is only partially solved even for one-sided queuing systems: Guo and Zipkin (2007) provides a partial characterization by finding sufficient conditions with respect to the patience distribution function. While our model considers three information regimes, it would be useful to expand the space of regimes to find the structure of the "optimal" information regime. Finally, another interesting extension would be to study service disciplines other than first-come-first-served (for example, a random service discipline), to better suit particular applications (for example, ride sharing).

Chapter 3

Dynamic Balancing of Efficiency and Risk in Multi-Class Screening Systems

3.1 Introduction

Security screening systems at land borders, airports, seaports and other military and civil facilities aim to identify malevolent people and illicit goods to mitigate downstream risks. The global scale of these operations is enormous. For example, an estimated 1.1 million passengers cross US borders every day, resulting in the seizure of around 8,000 drugs, $300,000 worth of currency and $3.8 million worth of products in violation of Intellectual Property Rights laws on a typical day (US Customs and Border Protection, 2016). At the same time, screening operations often create long queues at checkpoints, which impose significant system-wide costs.[1] Screening systems thus need to balance the objectives of risk management—i.e., ensuring reliable operations to identify threats—and operating efficiency—i.e., maintaining appropriate screening speeds to avoid long wait times.

This problem falls under the broad umbrella of speed-quality trade-offs in service operations, where human operators dynamically adjust service levels to ensure quality (through service slowdowns) while minimizing wait times (through service speedups). Screening systems involve similar dynamics, as operators solve the efficiency-risk trade-off by dynamically selecting the intensity of the screening procedure—by applying, for instance, a simple document check vs. a full body search. However, two unique features need to be considered in screening systems management: (i) the heterogeneity of people and goods being screened, and (ii) the strategic behaviors of agents when *people* (not goods) are subject to screening procedures.

First, all people and goods do not pose the same threat. For instance, many travelers have no intention of harm and carry appropriate travel documents; others may carry illicit items ranging from agricultural products to drugs; a few travelers might also pose extremely high threats by

[1] See, e.g., http://www.dispatch.com/news/20180609/legal-immigrants-can-wait-weeks-to-cross-us-mexico-border; http://time.com/4349766/airline-70000-passengers-missed-flights-due-to-security-lines/

bringing firearms or planning terrorist attacks. Operators obviously cannot identify benevolent vs. malevolent agents with certainty before applying screening procedures. Yet, operators often have access to some information on threat levels from initial screenings or from formal pre-screening programs developed to overcome information deficiencies—such as the Automated Targeting System and Secure Flight programs in the United States. This heterogeneity provides opportunities for operators to differentiate screening procedures based on pre-screening information. For instance, speeding up service for high probability benevolent agents, even when the system is near empty, can mitigate future wait times and avoid situations where service speedups are made necessary by long queues—with resulting high risk levels if malevolent agents benefit from speedups.

Second, many systems deal with the screening of human agents. Unlike packages, people can observe the screening procedures being applied and make strategic decisions regarding whether to go through the screening process or not. For instance, strict screenings may add small travel delays for benevolent agents but have more severe financial or penal consequences for malevolent agents, who may thus decide to leave the system prior to screening. From the perspective of the operators, the selection of screening procedures can be made more effective by anticipating these endogenous decisions from the agents. However, a major challenge lies in information asymmetries between the agents—who know their intentions with accuracy—and screening operators—who, despite some pre-screening information, can only assess agent types with uncertainty.

This chapter formulates decision-making models to select screening procedures in multi-class settings. It takes the viewpoint of a screening operator who must balance efficiency and risk by selecting one procedure for each job (person or package) from a menu of available ones. We first consider instances with non-strategic agents, to determine how the selection of screening procedures can leverage real-time information on queue lengths and each job's perceived threat level. We then extend our model to determine how anticipating agents' strategic decisions to go through screening or renege may impact the optimal selection of screening procedures. In both instances, we identify the value of pre-screening information with respect to expected queue lengths and risk costs. Before proceeding further, we provide three examples of multi-class screening systems with different screening procedures, job heterogeneity, pre-screening information and various extents of strategic behaviors:

Package and container screening: Up to 1 million bags and 75,000 containers are inspected per day with large-scale x-ray machines, gamma ray machines and radiation detection devices. Item-level information can be obtained from such characteristics as size, weight and point of origin. The system balances efficiency and risk in the presence of non-human, hence non-strategic agents.

Airport security: Airport security checkpoints are a major bottleneck of travel operations. Screenings are performed with x-ray machines, full body scanners, and human pat-downs. Pre-screening information stems from such programs as the Secure Flight and TSA Precheck in the United States. Here, only a tiny portion of malevolent agents are expected to pose high threats and to react

strategically to screening procedures.

Border Crossings: Screening procedures range from document checks to body searches, vehicle searches and canine searches. Agent heterogeneity stems from the array of possible custom laws violations, from minor offenses (e.g., carrying undeclared merchandise) to major ones (e.g., carrying drugs or firearms). Here, a higher proportion of travelers may exhibit strategic behaviors.

This chapter makes the following contributions:

- *Developing a dynamic decision-making model to balance efficiency and risk in multi-class queuing systems (Section 3.3).* We consider an $M/M/1$ queue with heterogeneous jobs. The operator controls the service rate by selecting the screening procedure applied to each job. We formulate the model as a continuous-time infinite-horizon Markov decision process that optimizes these decisions as a function of observed queue lengths and the threat level of each job. We show that the optimal policy exhibits a double threshold behavior: All else equal, the shorter the queue length and/or the higher the risk level, the stricter the optimal screening.

- *Assessing the impact of pre-screening risk information on system performance (Section 3.4).* We compare the optimal multi-class policy (with heterogeneous risk profiles) to a baseline single-class decision-making scheme based on queue length alone. Leveraging job-level risk information results in a Pareto improvement by mitigating both risk costs and queuing costs, with reductions in total expected costs of up to 6-7%. These savings increase with the granularity of risk profiles (i.e., the number of classes) and with the flexibility granted to screening operators (i.e., the number of available screening procedures).

- *Extending the model to account for agents' strategic behaviors (Section 3.5).* When the system deals with the screening of people (as opposed to goods), each agent can renege from the system prior to screening if the estimated risk of being flagged exceeds the benefits of successfully going through security. The model captures information asymmetries between agents and the screening operator. We formulate the operator's problem as a Markov decision process that incorporates the endogenous strategic behaviors of heterogeneous agents.

- *Assessing the impact of strategic behaviors and pre-screening information on optimal screening policies and system performance (Section 3.6).* Anticipating agents' strategic behaviors results in stricter screening to try to deliberately force malevolent agents to renege. By reducing service rates, the operator mitigates expected risks and may also, surprisingly, reduce expected queue lengths when the proportion of malevolent agents is significant—thus achieving a Pareto improvement. Moreover, better pre-screening information reduces total expected costs by mitigating information asymmetries; however it does not necessarily reduce expected risks.

From a practical standpoint, this chapter highlights the value of differentiating screening procedures over heterogeneous agents distinguished by pre-screening information, and provides decision-

making tools to support these decisions. Specifically, results underscore two benefits of pre-screening information: (i) characterizing agent heterogeneity through multi-class job profiles, and (ii) reducing information asymmetries between agents and system operators. As a result, pre-screening information can mitigate security risks *and* system congestion. At the same time, pre-screening information stems primarily from profiling programs, which raise broader questions of potential biases and privacy implications. This chapter contributes to this discussion by providing a transparent assessment of the operational impacts of pre-screening information.

3.2 Literature Review

This chapter bridges three streams of research: (i) security operations, (ii) quality-speed trade-offs in service operations, and (iii) strategic queuing. Specifically, it builds upon the security operations literature to optimize the selection of screening procedures, but extends it to capture the trade-off between risk and queuing costs and the impact of strategic customers.

Security Operations. Extensive research has focused on security systems at airport and seaport management, border control, counter-terrorism, etc. This literature falls into two broad categories. The first category designs screening procedures to mitigate security risks and operating costs. Several studies employ cost-benefit analyses to assess the effectiveness of screening procedures and pre-screening profiling programs (Barnett, 2004; Wein et al., 2006; Bakır, 2008; Cavusoglu et al., 2010). Researchers have also used security games to design defense strategies and detect malevolent agents, given attackers' strategic incentives (Sinha et al., 2016; Wrzaczek et al., 2017; Sinha et al., 2018). At the operational level, several studies aim to optimize the assignment of screening procedures to travelers, which is also focus of this chapter. McLay et al. (2007) and Nikolaev et al. (2007) propose integer programming models to optimize the screening of passengers and carry-on bags at airports, when passengers are heterogeneous in their risk profiles. McLay et al. (2010) and Lee and Jacobson (2012) address a similar problem when heterogenous passengers arrive sequentially, and formulate it as a stochastic Markov decision process.

The second category of research employs queuing models to estimate the impact of various screening strategies on system operations. Gilliam (1979) applies queuing theory to design passenger screening facilities at a large international airport. Atkinson and Wein (2008) use a spatial queuing model to shield cities from terrorist attacks with detection-interdiction systems. Zhang (2009) propose a congestion-based staffing policy at border crossings. Zhang et al. (2011) develop a two-stage inspection procedure, by optimizing the proportion of travelers subject to secondary screening and then the time spent in secondary screening.

This chapter bridges these two categories by dynamically selecting screening procedures applied to heterogeneous agents, while balancing risk and congestion mitigation in a queuing setting.

Speed-Quality Trade-Off. Service operations often involve a trade-off between wait times and service levels (Gans et al., 2003; Aksin et al., 2007). This literature considers queuing systems where the operator exercises dynamic control (Hopp et al., 2007a), such as systems of experts (Debo et al., 2008) or case managers (Campello et al., 2016). Our work relates to the literature on the control of service rates in queuing systems, balancing the operational benefits of shorter queue lengths vs. the implementation costs of faster service—in our case, the costs of higher security risks. Stidham and Weber (1989) and George and Harrison (2001) show that the optimal policy is monotonic, i.e., optimal service speeds increase as the queue gets longer. This has been extended to incorporate congestion pricing decisions (Ata and Shneorson, 2006), continuous decisions to keep working on a job or to stop (Hopp et al., 2007b), and returning customers when service quality is low (Chan et al., 2014). Our work extends this body of work by incorporating the heterogeneity of risk profiles and strategic agents in a multi-class setting.

Strategic Queuing. Naor (1969) proposes the first model of strategic queuing, in which customers trade off prices and wait times. Hassin and Haviv (2003) and Hassin (2016) survey the subsequent literature on this topic. Closely related to our setting, Anand et al. (2011) analyze the quality-speed trade-off in a strategic queuing setting, where the firm optimizes its pricing and service rates and customers make purchasing decisions based on prices, wait times and service quality. Wang et al. (2010) study a similar problem in the context of health diagnostic centers. They optimize staffing levels and a "certainty threshold" that nurses need to reach about patient pathologies before completing the diagnostic, given patients' strategic decisions when joining the system. Alizamir et al. (2013) incorporate Bayesian learning into the optimization of the "certainty threshold". In call centers, Zhan and Ward (2013) solve the problem between wait times and call resolution by optimizing the routing of calls when operators are heterogenous.

In Sections 3.5 and 3.6, we follow a similar approach by integrating agents' strategic decisions to remain in the system or renege into the selection of screening procedures. Two differences with the aforementioned studies are that (i) we model strategic behaviors in terms of reneging (i.e., agents leaving the system after joining) rather than balking (i.e., agents deciding not to join the system), and (ii) the operator's decisions are made dynamically for each job rather than statically.

3.3 Selection of Screening Procedures with Non-Strategic Agents

We propose an analytical model to dynamically optimize the selection of screening procedures in a setting with multiple classes of jobs and non-strategic agents. Job heterogeneity reflects the various risk levels, as assessed by the operators from pre-screening information. Non-strategic agents may include items such as packages, bags and containers, which do not adjust their behaviors in response to the screening procedure applied by the operator. In Section 3.5, we relax this assumption by considering human agents' strategic behaviors.

We formulate the model in Section 3.3.1. We then characterize the optimal policy in Section 3.3.2, and show that it exhibits a double threshold behavior. Specifically, the longer the queue and/or the lower the risk level of the job under consideration, the faster the optimal screening procedure.

3.3.1 Model Development

We formulate the problem as a continuous-time Markov decision process. We consider a first-come first-served $M/M/1$ queuing system.[2] Screening jobs arrive according to an exogenous Poisson process, and service is provided by one operator. For each job, the operator selects one out of several screening procedures. Procedures vary with respect to service speeds and risk costs: More conservative procedures (e.g., full body search) induce a lower service speed but a lower risk cost than more expedited ones (e.g., document check). Therefore, the operator controls both the service rate of the queue and the risk costs faced by the system.

The operator selects the screening procedure based on the queue length and the perceived risk level of the job under consideration. Queue length information can be obtained in real-time by counting the number of jobs in queue or through sensors. Risk levels are available from pre-screening programs and any relevant real-time observation of the operator. We assume that operators assess the risk level of jobs when they reach service.

We introduce the following notation to describe job arrivals, risk heterogeneity, and screening procedures. Let λ denote the arrival rate of jobs. The perceived risk level of each job—observable by the screening operator—is denoted by $r \in \Re+$; higher values of r reflect higher risk levels. We assume that the risk level r is sampled from a known probability distribution function, denoted by f. For each job, the operator chooses a screening procedure $\phi \in \Theta$, where Θ denotes a set of available procedures (e.g., document check, x-ray scanning, full body search). For each procedure ϕ, the service time follows an exponential distribution, whose rate is denoted by μ_ϕ.

We denote by $R_\phi(r)$ the risk cost resulting from screening a job with a risk level $r \in \Re+$ using screening option $\phi \in \Theta$. The risk costs can be interpreted as the expected cost of false negatives, i.e., the expected security costs resulting from a malevolent item not being detected by the system.[3] Specifically, we denote by p_ϕ the probability of a false negative when procedure ϕ is applied, and by $C(r)$ the cost of a false negative for a job with risk level r. Note that we assume, for analytical tractability, that the probability of false negatives is independent of the risk profiles. We have:

$$R_\phi(r) = p_\phi C(r), \quad \forall r \in \Re+, \forall \phi \in \Theta. \tag{3.1}$$

We make the following assumptions to reflect the trade-off between efficiency and risk:

Assumption 1. *For any $\phi, \psi \in \Theta$: If $\mu_\psi \geq \mu_\phi$, then $R_\psi(r) \geq R_\phi(r), \forall r \in \Re+$.*

[2] Our insights can hold in more general settings with two or more operations. However, we consider an $M/M/1$ system for analytical tractability. This is consistent with the literature (Ata and Shneorson, 2006).

[3] We assume that the cost of false positives (i.e. increased delay of benevolent jobs) is reflected in increased delay.

Assumption 2. *For any $r_1, r_2 \in \Re+$ such that $r_1 \leq r_2$: $R_\phi(r_1) \leq R_\phi(r_2), \forall \phi \in \Theta$.*

These assumptions are motivated as follows. First, the faster a screening procedure, the lower its reliability and, hence, the larger the probability of false negatives. In other words, for each $\phi, \psi \in \Theta$ such that $\mu_\phi \leq \mu_\psi$, we have: $p_\phi \leq p_\psi$.[4] This assumption is consistent with the literature on the quality-speed trade-off, where higher speeds of service induce lower quality. Second, the higher the risk level of a job, the larger the security costs in case of a false negative. In other words, for each $r_1, r_2 \in \Re+$ such that $r_1 \leq r_2$, we have: $C(r_1) \leq C(r_2)$. This assumption is consistent with security operations literature where higher costs are associated with more risky profiles. Assumptions 1 and 2 provide more general conditions than these statements.

The operator selects a screening procedure each time an event occurs, i.e., each time a new service is initiated, or a new job arrives.[5] This is formulated as a continuous-time infinite-horizon Markov decision process. Let t_k denote the time at which the k^{th} event (job arrival or service completion) occurs. The state variable characterizes the observed number of jobs in the system (i.e., in queue and in service), denoted by i, and, if $i \geq 1$, the risk profile of the job being screened, denoted by r. Therefore, the state space of the problem $\mathcal{S}$ is given by:

$$\mathcal{S} = \{0\} \times \{(i, r), i \geq 1, r \in \Re+\}. \tag{3.2}$$

At each epoch t_k, the decision space includes the elements of the set Θ if $i \geq 1$; no decision is made if $i = 0$ (i.e., the system is empty). Let i_k be the number of jobs in the system at time t_k; let r_k be the risk level of the next agent at time t_k, if $i_k \geq 1$; and let ϕ_k be the screening procedure selected at time t_k, if $i_k \geq 1$. We denote by π a policy that associates an element of Θ to any element of $\mathcal{S}$.

The cost function comprises two terms: (i) a queuing cost accrued continuously over time, and (ii) the risk cost, incurred upon completion of each service. We denote the queuing cost per unit of time by $g(i)$. The risk cost is equal to $R_{\phi_k}(r_k)$ if the next transition is a service completion and 0 otherwise. We denote by δ_k a Bernoulli variable that is equal to 1 if the transition from t_k to t_{k+1} is a service completion, and 0 otherwise. Let $\beta > 0$ be the operator's continuous discount rate. The model's objective function, denoted by z, is thus given by:

$$z = E\left[\sum_{k=0}^{\infty}\left(\int_{t_k}^{t_{k+1}} g(i_k)e^{-\beta t}dt + e^{-\beta t_{k+1}} R_{\phi_k}(r_k)\delta_k\right)\right]. \tag{3.3}$$

The transition rates are non-uniform, i.e., the time interval between two consecutive transition epochs depends on the operator's screening decisions. Indeed, transitions follow a Poisson process with rate $\lambda + \mu_{\phi_k}$ when the system is non-empty, and with rate λ when the system is empty. We apply the method of *uniformization* (Bertsekas, 2005, 2012): Let $\bar{\mu} = \max\{\mu_\phi, \phi \in \Theta\}$ denote the fastest

[4] Any service procedure for ϕ such that $\mu_\phi \leq \mu_\psi$ and $p_\phi > p_\psi$ can be removed from the set under consideration.

[5] Note that this enables the operator to change its procedure during an inspection if a new job arrives. Practically there is nothing to prevent this if the jobs being considered are goods.

screening rate among all available procedures. Then $\lambda + \bar{\mu}$ provides an upper bound of the transition rate from each state and under any policy π. We consider a modified Markov decision process, where the rate of each transition is equal to $\lambda + \bar{\mu}$ by adding self-transitions. Specifically, we add self-transitions with rate $\bar{\mu} - \mu_{\pi(i,r)}$ in each state (i, r) and with rate $\bar{\mu}$ in state 0. The corresponding state transition probabilities in any state $s \in \mathcal{S}$ under policy π, denoted by $(P_{s\to s'}(\pi))_{s'\in\mathcal{S}}$, are given as follows, where $\mathbb{1}$ denotes the indicator function:

$$
\begin{array}{rclll}
P_{(i,r_0)\to(i+1,r)}(\pi) & = & \frac{\lambda}{\bar{\mu}+\lambda}\mathbb{1}(r=r_0) & \forall i \geq 1, r_0, r \in \Re+ \\
P_{(i,r_0)\to(i,r)}(\pi) & = & \frac{\bar{\mu}-\mu_{\pi(i,r_0)}}{\bar{\mu}+\lambda}\mathbb{1}(r=r_0) & \forall i \geq 1, r_0, r \in \Re+ \\
P_{(i,r_0)\to(i-1,r)}(\pi) & = & \frac{\mu_{\pi(i,r_0)}}{\bar{\mu}+\lambda} f(r) & \forall i \geq 2, r_0, r \in \Re+
\end{array}
\qquad
\begin{array}{rclll}
P_{0\to(1,r)}(\pi) & = & \frac{\lambda}{\bar{\mu}+\lambda} f(r) & \forall r \in \Re+ \\
P_{0\to 0}(\pi) & = & \frac{\bar{\mu}}{\bar{\mu}+\lambda} & \\
P_{(1,r_0)\to 0}(\pi) & = & \frac{\mu_{\pi(1,r_0)}}{\bar{\mu}+\lambda} & \forall r_0 \in \Re+
\end{array}
$$

Figure 3.1 illustrates the transition diagram of the system under policy π with two risk levels denoted by r_1 and r_2, each occurring with probability h_1 and $h_2 = 1 - h_1$, respectively.

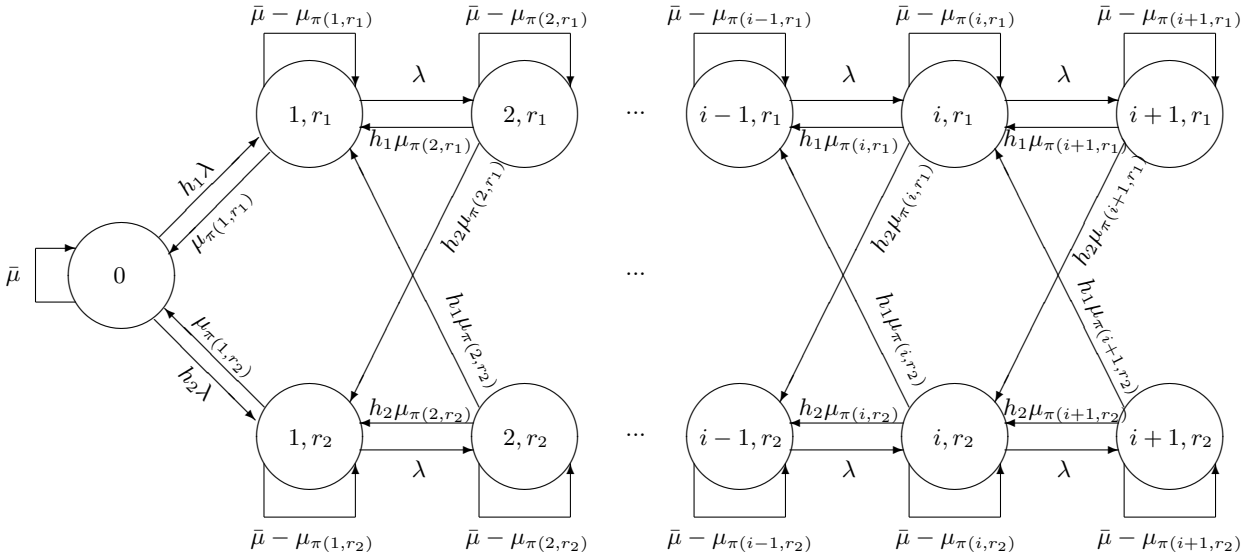


Figure 3.1: State-transition diagram under policy π.

Based on the problem's re-formulation as a continuous Markov decision process with uniform transition rates, we derive the discrete and recursive formulation of the Bellman equation in Lemma 3.1, by decomposing the cost function from one decision epoch to the next. The proof follows the procedure from Bertsekas (2005, 2012), and thus omitted for conciseness.

Lemma 3.1. *Let $J(i, r)$ denote the cost-to-go function for the problem. Let $\boldsymbol{J}(i)$ be the expected cost-to-go across all risk profiles when i jobs are present in the system, i.e.:*

$$\boldsymbol{J}(0) = J(0) \qquad \textit{and} \qquad \boldsymbol{J}(i) = \mathbb{E}_r[J(i,r)] = \int_{r\in\Re+} J(i,r) f(r) dr, \quad \forall i \geq 1. \tag{3.4}$$

The Bellman equation of the problem is given as follows:

$$J(i,r) = \frac{1}{\beta+\bar{\mu}+\lambda}\Bigg(g(i) + \lambda J(i+1,r) + \min_{\phi\in\Theta}\left\{\mu_\phi R_\phi(r) + \mu_\phi \boldsymbol{J}(i-1) + (\bar{\mu}-\mu_\phi)\, J(i,r)\right\}\Bigg) \qquad \forall i \geq 1, \forall r \in \Re+, \tag{3.5}$$

$$J(0) = \frac{1}{\beta+\bar{\mu}+\lambda}\left(g(0) + \lambda \boldsymbol{J}(1) + \bar{\mu} J(0)\right). \tag{3.6}$$

The Bellman equation comprises five terms. First, $\frac{g(i)}{\beta+\bar{\mu}+\lambda}$ captures the discounted queuing cost, incurred continuously between two consecutive decision epochs. Second, $\frac{\lambda}{\beta+\bar{\mu}+\lambda}J(i+1,r)$ refers to the discounted cost-to-go following the arrival of the next job. Next, $\frac{\mu_\phi}{\beta+\bar{\mu}+\lambda}R_\phi(r)$ and $\frac{\mu_\phi}{\beta+\bar{\mu}+\lambda}\mathbf{J}(i-1)$ represent the discounted risk cost and cost-to-go following a job completion. The cost-to-go function is averaged over all risk levels, reflecting the uncertainty regarding the risk level of the next job. Last, $\frac{\bar{\mu}-\mu_\phi}{\beta+\bar{\mu}+\lambda}J(i,r)$ reflects self-transitions.

We denote the optimal cost-to-go function by J^* and the optimal policy by π^*. Also, let $\mathbf{J}^*(i)$ be the optimal value of $\mathbf{J}(i)$. From Equation (3.5), we obtain that for each $i \geq 1$ and each $r \in \Re+$:

$$\pi^*(i,r) = \phi \iff (\mu_\phi - \mu_\psi)\left[J^*(i,r) - \mathbf{J}^*(i-1)\right] \geq \mu_\phi R_\phi(r) - \mu_\psi R_\psi(r), \quad \forall \psi \in \Theta. \tag{3.7}$$

3.3.2 Characterization of Optimal Policy

We now characterize the optimal policy as a function of the observed queue length $i \geq 1$ and risk level $r \in \Re+$. To simplify the exposition, we assume that $|\Theta| = 2$. In other words, the operator can select one of two screening procedures: "fast" and "slow" which we index by "F" and "S", respectively. We denote by μ_F and μ_S the corresponding screening rates. We have $\mu_F \geq \mu_S$, so $\bar{\mu} = \mu_F$. For any risk level r, the corresponding risk costs are denoted by $R_F(r)$ and $R_S(r)$. Computational results in Section 3.4.1 indicate that all our results hold with three or more screening procedures.

Due to (3.7) the optimal policy is given by:

$$\pi^*(i,r) = \text{"fast"} \iff J(i,r) - \mathbf{J}(i-1) \geq \frac{\mu_F R_F(r) - \mu_S R_S(r)}{\mu_F - \mu_S}.$$

We now introduce additional structural assumptions on the queuing and risk costs:

Assumption 3. *The waiting cost function $g(i)$ is non-decreasing and convex in i. This is a standard assumption in the literature, also supported by empirical evidence (Van Mieghem, 1995).*

For any risk level r, the risk under fast service is larger than the risk under slow service, i.e., $R_F(r) \geq R_S(r)$ for all $r \in \Re+$ (Assumption 1); and the risk costs $R_F(r)$ and $R_S(r)$ are both increasing in r (Assumption 2). In addition, we impose Assumptions 4 and 5.

Assumption 4. $R_S(r) \leq \frac{g(1)-g(0)}{\beta}, \quad \forall r \in \Re+.$

Assumption 5. $\frac{\mu_F}{\mu_F+\beta}\left(R_F(r_2)-R_F(r_1)\right) \geq \frac{\mu_S}{\mu_S+\beta}\left(R_S(r_2)-R_S(r_1)\right) \quad \forall r_1 < r_2 \in \Re+.$

Assumption 4 ensures that whenever there is a single job of any risk level r in the system, the screening operator will always process it even with the slowest option and incur cost $R_S(r)$, rather than keeping the job waiting indefinitely. In other words, the slowest screening procedure is always preferable to no screening at all. Note that Assumption 4 is satisfied with β sufficiently small. Next, as β nears 0, Assumption 5 can be written as $R_F(r_2) - R_F(r_1) \geq R_S(r_2) - R_S(r_1)$ for all $r_1 < r_2 \in \Re+$: The marginal increase in risk costs with high and low risk levels is larger under fast screening than under slow screening. Equivalently, $R_F(r_2) - R_S(r_2) \geq R_F(r_1) - R_S(r_1)$ for all $r_1 < r_2 \in \Re+$: The marginal increase in risk costs between slow and fast screening is higher for riskier profiles than less risky ones. Assumption 5 provides a more relaxed condition as $\beta > 0$.

We now show that the optimal policy exhibits a *double threshold behavior* in queue lengths and risk levels in Propositions 3.2 and 3.3. First, Proposition 3.2 shows that the optimal policy is monotonic in queue lengths: all else equal, the longer the queue length, the faster the screening. In other words, for any risk level r, there exists a value $\hat{i}_r$ such that the "fast" procedure is employed when $i \geq \hat{i}_r$, and the "slow" procedure is employed when $i < \hat{i}_r$. This result extends Stidham and Weber (1989) and George and Harrison (2001) to a multi-class setting with heterogeneous jobs.

Proposition 3.2. *The optimal policy π^* exhibits the following threshold pattern: For any risk level $r \in \Re+$ and any queue length $i \geq 0$, we have:*

$$\text{If } \pi(i,r) = \mu_F \text{ then } \pi(i+1,r_1) = \mu_F. \tag{3.8}$$

Second, Proposition 3.3 shows that the optimal policy is monotonic in the risk level: all else equal, the riskier the job, the slower the screening. In other words, for any queue length i, there exists a value $\hat{r}_i$ such that the "fast" procedure is employed when $r \leq \hat{r}_i$, and the "slow" procedure is employed when $r > \hat{r}_i$. This threshold behavior further extends the literature on the speed-quality trade-off to multi-class settings by underscoring the dependencies of the screening policy on available risk information. It also suggests potential operational benefits resulting from differentiating the screening policies with respect to risk profile information. Specifically, relaxing the screening requirements for low-risk jobs—even when the system is near empty—can enable system operators to apply stricter screening for higher-risk jobs later. This is further discussed in Section 3.4.1. Together, Propositions 3.2 and 3.3 provide decision-making support regarding when to switch from a given screening procedure to an alternative one by considering both operational factors (e.g., queue length, service rates) and risk factors (e.g., risk profile information, risk costs).

Proposition 3.3. *The optimal policy π^* exhibits the following threshold pattern: For any queue length $i \geq 0$ and risk levels $r_1, r_2 \in \Re+$ such that $r_1 \leq r_2$, we have:*

$$\text{If } \pi(i,r_2) = \mu_F \text{ then } \pi(i,r_1) = \mu_F. \tag{3.9}$$

Propositions 3.2 and 3.3 are proved by induction using the value iteration algorithm, by showing, respectively, that $\frac{\mu_F - \mu_S}{\beta+\lambda+\mu_F} J(i,r) - R_F(,r) + R_S(r)$ is decreasing in r for any value of i, and that $J(i,r) - \mathbf{J}(i-1)$ is increasing in i for any value of r. Both proofs are reported in Appendix C.2.

3.4 Computational Results

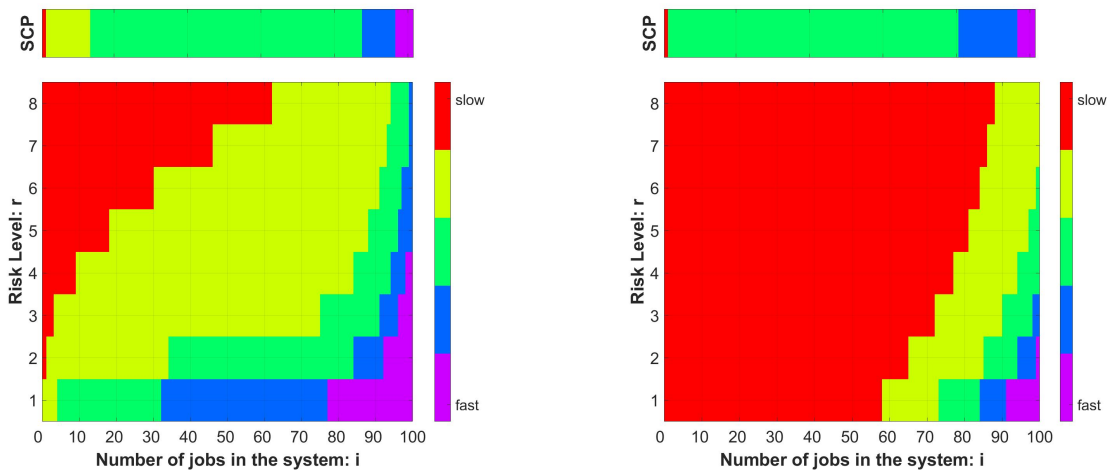
We now show computational results to illustrate the optimal policy and extend our analytical insights to a setting with three or more screening options (Section 3.4.1), and to identify the benefits of the multi-class model developed in this chapter, vs. existing single-class models (Section 3.4.2). We obtain the optimal policy by applying a value iteration algorithm (Bertsekas, 2005, 2012; Powell, 2007), with a finite state space approximation obtained by placing an upper bound on the queue length. The full experimental setup is detailed in Appendix C.1.

We refer to the optimal policy as *multi-class policy*, and to a baseline policy that does not leverage risk information as *single-class policy*. The single-class policy is obtained by solving Equations (3.5) and (3.6) with a one-dimensional state i representing the number of jobs in the system, i.e., by selecting the screening procedure for each value of i regardless of the current job's risk level.

3.4.1 Optimal Policy

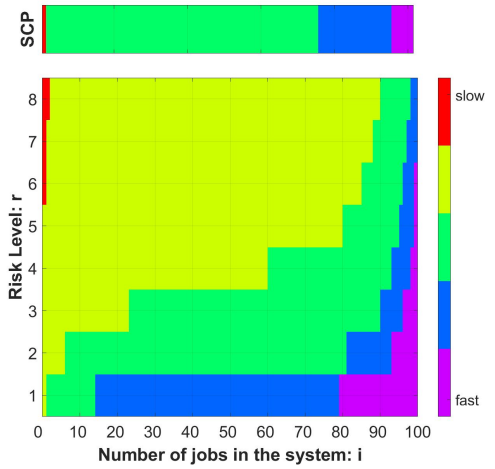
Figure 3.2 shows the optimal multi-class and single-class policies as a function of the observed queue length and risk level. We consider here five screening options, and a discrete risk distribution with eight risk profiles. The optimal policies are indicated by the color of the region, with red (resp. purple) representing the slowest (resp. fastest) screening option. The single-class policy ("SCP") is shown at the top of each graph as a function of the number of jobs in the system; the multi-class policy is shown at the bottom of each graph in a two-dimensional setting representing the number of jobs in the system and the risk level of a job. Figure 3.2a shows a baseline case to illustrate the optimal policy in a two-dimensional state space. Figure 3.2b, 3.2c and 3.2d then show the optimal policy with higher risk costs, greater spread of screening rates across the five available procedures and higher arrival rate, respectively. Higher risk costs are modeled by multiplying all values of $R_\phi(r)$ by a constant factor; greater spread of screening rates are obtained by increasing the difference between the largest and the slowest screening rates, while keeping the error rates constant and distributing the intermediate ones uniformly between the two extremes.

Note the threshold behavior of the optimal multi-class policy with respect to both the queue length and the risk level: the larger the risk level, the lower the service rate; the larger the number of jobs in the system, the larger the service rate. This is consistent with our analytical findings (Propositions 3.2 and 3.3), and extends them to a setting with three or more available procedures. Moreover, by differentiating the screening procedure based on risk levels, the multi-class policy results in stricter screening for higher-risk jobs and looser screening for lower-risk jobs than the

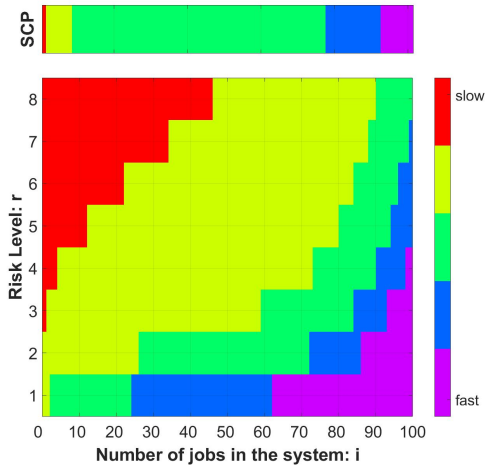


(a) Base case

(b) Higher risk costs



(c) High spread of service rates



(d) Higher demand

Figure 3.2: Single-class and multi-class policies for different service speeds, risk profiles and job demand.

single-class policy. For example, in Figure 3.2a, when the system is near empty, the single-class policy applies the slowest procedure, while the multi-class policy speeds up operations for low-risk jobs. In contrast, under higher congestion (e.g., $i = 40$ or $i = 90$), the multi-class policy applies strict screenings for high-risk jobs (shown in red and yellow), while the single-class policy applies faster procedures (shown in green and blue). As we shall discuss next, such differentiation in the multi-class policy reduces expected queue lengths (through speedups for low-risk jobs) as well as risk costs (through slowdowns for high-risk jobs).

The figures also show the sensitivity of the optimal policy with respect to the model parameters. First, slower screening is applied as the risk costs increase (Figures (3.2a) vs (3.2b)): Higher risk costs require more conservative screening for risk mitigation, at the expense of higher queuing

costs. Second, faster screening is applied as the spread of the service rates increases (Figures (3.2a) and (3.2c)): More differentiated screening options increase the marginal benefits of switching from slower to faster screening for queue length mitigation, at the expense of higher risk costs. Third, faster screening is applied as the arrival rate increases (Figures (3.2a) and (3.2d)): higher demand increases future expected queuing costs, hence motivating faster screening at the expense of higher risk costs. These insights underscore the interplay between the two objectives of efficiency and risk in screening systems, and the ways operators can trade off these two objectives by adjusting the queue-based and risk-based thresholds, as a function of the operating environment.

3.4.2 Benefits of Multi-class Risk Information and Screening Flexibility

We now quantify the benefits of accounting for heterogeneous risk profiles when several screening procedures are available: Several available screening options provides operating flexibility to apply differentiated procedures based on observed queue lengths (typical in service operations facing a quality-speed trade-off) and risk levels (a new dimension introduced here). We thus evaluate here the *joint* benefits of risk information—captured by the multiple classes of jobs—and screening flexibility—captured by the availability of several screening options.

To this end, we report, for the multi-class policy and the single-class policy, the value of the cost-to-go function in state 0, that is $J(0)$. This can be interpreted as the total discounted cost starting from the beginning of a business day. We further decompose the cost-to-go function into risk cost and queuing cost, denoted by J_R and J_Q, respectively. They are defined as follows:

$$J_R(i,r) = \frac{\lambda J_R(i+1,r) + \mu_{\pi^*(i,r)} R_{\pi^*(i,r)}(r) + \mu_{\pi^*(i,r)} \mathbf{J}_R(i-1) + \left(\bar{\mu} - \mu_{\pi^*(i,r)}\right) J_R(i,r)}{\beta + \bar{\mu} + \lambda} \quad \forall i \geq 1, \forall r \in \Re+, \tag{3.10}$$

$$J_Q(i,r) = \frac{g(i) + \lambda J_Q(i+1,r) + \mu_{\pi^*(i,r)} \mathbf{J}_Q(i-1) + \left(\bar{\mu} - \mu_{\pi^*(i,r)}\right) J_Q(i,r)}{\beta + \bar{\mu} + \lambda} \quad \forall i \geq 1, \forall r \in \Re+, \tag{3.11}$$

$$J_R(0) = \frac{\lambda \mathbf{J}_R(1) + \bar{\mu} J_R(0)}{\beta + \bar{\mu} + \lambda}, \qquad J_Q(0) = \frac{g(0) + \lambda \mathbf{J}_Q(1) + \bar{\mu} J_Q(0)}{\beta + \bar{\mu} + \lambda}, \tag{3.12}$$

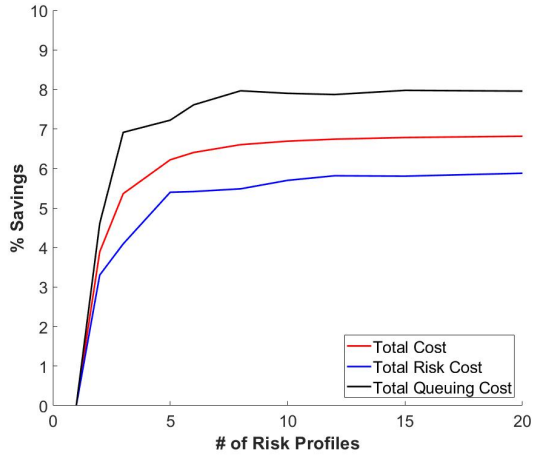
where $\mathbf{J}_R(i)$ and $\mathbf{J}_Q(i)$ denote the average values of J_R and J_Q in state i, respectively, over all risk levels $r \in \Re+$. By construction, we have: $J = J_R + J_Q$. We use the superscripts "S" and "M" to refer to the single-class policy and the multi-class policy, respectively,

Figures 3.3a and 3.3b compare the performance of the multi-class policy to that of the single-class policy, for different numbers of risk profiles and screening options, respectively. They show the percent-wise reduction of the total cost, risk cost and queuing cost obtained with the multi-class policy, i.e.:

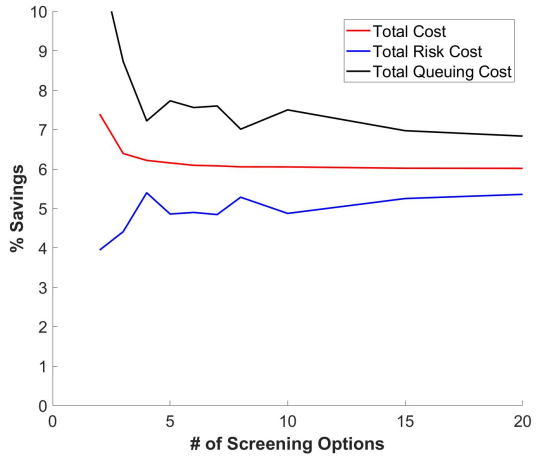
$$\frac{J^S(0) - J^M(0)}{J^S(0)}, \quad \frac{J_R^S(0) - J_R^M(0)}{J_R^S(0)}, \quad \text{and} \quad \frac{J_Q^S(0) - J_Q^M(0)}{J_Q^S(0)}.$$

Figures 3.3c and 3.3d show the total expected cost $J^M(0)$ for different numbers of risk profiles and screening options. The dashed lines correspond to $J^S(0)$, i.e., the baseline total cost from the single-class policy.

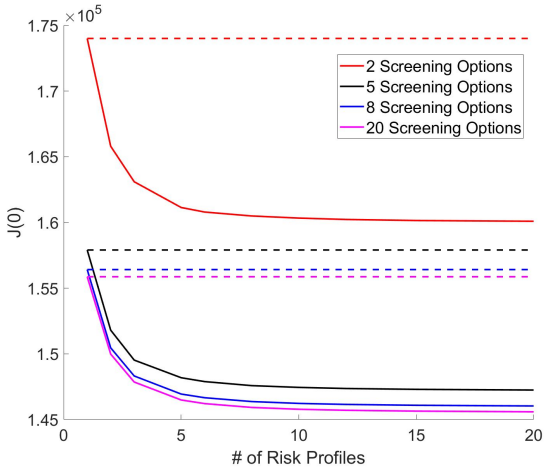
First, note from Figures 3.3a and 3.3b that the multi-class policy reduces total expected costs by up



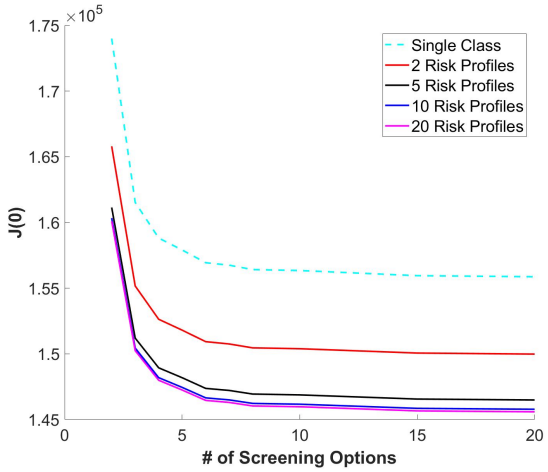
(a) Benefits of multi-class policy vs. # of risk profiles



(b) Benefits of multi-class policy vs. # of procedures



(c) Total discounted costs vs # of risk profiles



(d) Total discounted costs vs # of procedures

Figure 3.3: Joint benefits of risk information and screening flexibility.

to 6-7%, as compared to the single-class policy. Most importantly, the multi-class policy provides a Pareto improvement by reducing both risk and queuing costs: Leveraging risk information results in both safer operations *and* lower expected wait times. By applying faster screening procedures for low-risk jobs, especially under low congestion, the multi-class policy can prevent the formation of long queues, at a small increase in risk costs; but this provides opportunities to apply stricter procedures for high-risk jobs, even with longer queue lengths, which in turn decreases the risk costs faced by the system.

Next, Figures 3.3a and 3.3c show that total expected costs are further reduced as more job classes are considered, but at a diminishing rate. In our setting, total costs (as well as risk and queuing costs) reach a plateau with 5-10 risk profiles. This underscores the benefits of even limited risk information, as even non-granular classifications of jobs based on risk profiles can bridge a significant portion of the gap between the cost of the single-class policy, on the one hand, and the minimum cost that can be achieved with highly granular risk information, on the other. This suggests that even relatively unsophisticated pre-screening

programs could result in significant reductions in both total risk costs and queuing costs.

Similarly, Figures 3.3b and 3.3d show that total expected costs decrease with the number of available screening options and that returns are again diminishing. In other words, when only a few screening procedures are available, additional flexibility (i.e., a larger set of screening options) enables the operator to further differentiate screenings based on queue lengths and risk information. However, when many procedures are available, the marginal option has a more limited impact on optimal policies. Moreover, Figure 3.3b suggests that screening flexibility provides higher benefits in a single-class setting than in a multi-class setting, where risk information already enables granular selection of screening procedures by the operator.

Finally, Figures 3.3c and 3.3d underscore the *joint* benefits of risk information and screening flexibility. For any number of screening options, more granular risk information can reduce expected costs, up to a certain point. Further benefits can then be achieved only through additional flexibility, i.e., more available screening procedures. At one extreme, when only two screening options are available, the operator is limited in leveraging risk information due by constraints on screening procedures. The most significant benefits can thus be achieved by designing more screening procedures. However, more screening options can reduce costs up to a certain point in the absence of granular risk information. This underscores that the numbers of risk profiles and screening options have diminishing returns when considered separately, but can complement each other— yielding more significant benefits when considered together.

In conclusion, leveraging risk information can improve the performance of screening systems by discriminating screening policies based on risk profiles—in addition to queue lengths. Moreover, the multi-class policy proposed here yields a Pareto improvement by reducing both risk and queuing costs, as compared to a baseline single-class policy. Finally, our results underscore the interplay between risk information and screening flexibility, thereby suggesting opportunities for system planners to design screening procedures and pre-screening programs simultaneously based on implications for risk mitigation and operating efficiency.

3.5 Selection of Screening Procedures with Strategic Agents

We now extend the model presented in Section 3.3 to incorporate strategic behaviors of human agents subject to the screening procedures. We formulate the model in this section, and characterize the optimal policy and resulting system dynamics in the following one.

3.5.1 Model of Agents' Strategic Behaviors

We consider the following setting. Each agent ("she") joins the system, regardless of the queue length and the screening policy. Just prior to their service, however, each agent can deduce the procedure to be applied by the operator, based on the observed queue length and her own risk profile. Accordingly, each agent decides to remain in the system and go through screening, or to renege and leave before screening.

From the operator's perspective, accounting for each agent's endogenous decision to go through screening or renege may impact the optimal screening policy and resulting system performance. This problem is complicated by information asymmetries: Each agent knows her true type accurately but the operator can only assess it probabilistically (again, from pre-screening programs and any other observational information).

We denote by $s \in \Re+$ the *true profile* of an agent. The larger the s, the more malevolent the agent. We denote by $r \in \Re+$ the perceived *risk level* of an agent, as assessed by the system's operator. This is analogous to the variable r defined in Section 3.3; we still denote by f the (known) probability distribution of risk levels

r across the population. To capture information asymmetries, we consider a probabilistic mapping between r and s: let $q_r(s)$ be the probability distribution function of the agent's true profile, conditionally on being assigned a risk level r by the system's operator.

Assumption 6. *If Agents 1 and 2 are assigned risk levels r_1 and r_2, respectively, with $r_1 \leq r_2$, then Agent 2's true profile is more risky than Agent 1's. Mathematically, $q_{r_1}(\cdot)$ is stochastically dominated by $q_{r_2}(\cdot)$:*

$$\textit{For any } r_1, r_2 \in \Re+ \textit{ such that } r_1 \leq r_2, \textit{ for any } \bar{s} \in \Re+: \int_{s \geq \bar{s}} q_{r_2}(s)ds \geq \int_{s \geq \bar{s}} q_{r_1}(s)ds. \tag{3.13}$$

We denote by $K(s)$ the true risk cost resulting from an agent with true type $s \in \Re+$ going through the screening process successfully.

Assumption 7. *More risky profiles result in higher expected risk costs, i.e., $K(s)$ is increasing in s:*

$$\textit{For any } s_1, s_2 \in \Re+ \textit{ such that } s_1 \leq s_2\textit{: } K(s_1) \leq K(s_2). \tag{3.14}$$

The true risk cost $K(s)$ is related to the function $R_\phi(r)$ that was introduced in Section 3.3. Recall that $R_\phi(r)$ was defined as the probability of a false negative under screening procedure ϕ (i.e., p_ϕ) times the expected risk cost resulting from an agent of risk level r going through the screening process successfully (i.e., $C(r)$). The expected risk cost $C(r)$ is, itself, equal to the expected value of an agent's true risk costs, conditional on the agent being assigned to risk level r. Mathematically, we can express $R_\phi(r)$ as follows:

$$R_\phi(r) = p_\phi \int_{s \in \Re+} q_r(s)K(s)ds. \tag{3.15}$$

Note that, under assumptions (6) and (7), $R_\phi(r)$ satisfies Equation (2), i.e., $R_\phi(r)$ is non-decreasing in r.

Recall that p_ϕ denotes the probability of a false negative when procedure ϕ is employed. This probability is increasing in the speed of the screening procedure, i.e., for all $\phi, \psi \in \Theta$ such that $\mu_\phi \leq \mu_\psi$, we have: $p_\phi \leq p_\psi$. Accordingly, from the perspective of an agent, it can be interpreted as the probability of successful screening. Each agent of type $s \in \Re+$ decides to go through screening whenever the probability of successful screening exceeds a threshold denoted by τ^s, and reneges otherwise. The parameter τ^s captures the key differentiation between agent types. At one extreme, for a benevolent agent (with a low value of s) unsuccessful screening would result in a small travel delay, with low resulting costs. Vice versa, for a malevolent agent (with a higher value of s) unsuccessful screening may lead to more severe consequences (e.g., financial fine, legal prosecution). Following that logic, we posit that a benevolent agent will be willing to go through stricter screening procedures than malevolent agents. Mathematically, this translates into Assumption 8 that τ^s is non-decreasing in $s \in \Re+$, i.e.:

Assumption 8. *For any $s_1, s_2 \in \Re+$ such that $s_1 \geq s_2$: $\tau^{s_1} \geq \tau^{s_2}$.*

For each screening procedure $\phi \in \Theta$, we denote by S_ϕ^+ and S_ϕ^- the set of agents (characterized by their true profiles $s \in \Re+$) that will go through the system and renege, respectively:

$$S_\phi^+ := \{s \in \Re + | p_\phi > \tau^s\}, \tag{3.16}$$

$$S_\phi^- := \{s \in \Re + | p_\phi \leq \tau^s\}. \tag{3.17}$$

Note that the risk faced by the system is zero when an agent decides to renege prior to screening. Therefore, the effective risk cost faced by the system when procedure ϕ is selected, denoted by $R^S_\phi(r)$, is averaged over the risk profiles in S^+_ϕ only and may thus be lower than $R_\phi(r)$. Specifically, we have:

$$R^S_\phi(r) = p_\phi \int_{s \in S^+_\phi} q_r(s) K(s) ds. \tag{3.18}$$

3.5.2 Model Formulation

The selection of screening procedures with strategic agents involves the following dynamics: Each time an agent enters service, the operator selects a screening procedure; then, the agent decides to go through screening or to renege; if the agent proceeds to screening, the operator applies the selected procedure until service completion, and the process is repeated. Note that the operator cannot change the screening procedure during service (e.g., when a new job arrives into the queue)—reflecting the dynamics of screening and reneging with human agents. This differs from Section 3.3, where the focus was on non-human jobs (e.g., packages, containers), thus making it possible to change the screening procedure during service. This assumption does not impact the optimal policy and resulting insights significantly.

To capture the reneging dynamics, we consider the following state space: $\mathcal{S} = \{0\} \times \{(i,r), i \geq 1, r \in \Re+\} \times \{(i,r,\phi), i \geq 1, r \in \Re+, \phi \in \Theta\}$. Decisions are made only in any state in the set $\{(i,r), i \geq 1, r \in \Re+\}$; as in Section 3.3, state (i,r) signifies that $i \geq 1$ agents are present in the system and the first agent is of type $r \in \Re+$. If the operator selects screening procedure ϕ and the agent reneges (i.e., $s \in S^-_\phi$), then the system transitions (instantaneously) to some state $(i-1, r')$ (if $i \geq 1$), where r' denotes the risk level of the next agent and is sampled from distribution f; if $i = 1$, the system transitions to state 0. If the agent remains in the system (i.e., $s \in S^+_\phi$), in contrast, the system transitions (instantaneously) to state (i,r,ϕ). State (i,r,ϕ) indicates that $i \geq 1$ agents are in the system, that the next agent is of type $r \in \Re+$, that procedure $\phi \in \Theta$ is applied, and that the agent did not renege. Note that this state augmentation is necessary because, any time a service begins, the operator knows that the agent's true type is in S^+_ϕ and not in S^-_ϕ—so information in state (i,r,ϕ) is greater than in state (i,r).

For any $\phi \in \Theta$ and $r \in \Re+$, we denote by $w(\phi, r)$ the probability that an agent of risk level r will remain in the system if screening procedure ϕ is applied. It is given by:

$$w(\phi, r) = \int_{s \in S^+_\phi} q_r(s) ds. \tag{3.19}$$

We denote by $J^S(s)$ the cost-to-go function of the system with strategic agents, for state $s \in \mathcal{S}$. Let $\mathbf{J}^S(i)$ be the expected cost-to-go across all risk levels $r \in \Re+$ when i jobs are present in the system:

$$\mathbf{J}^S(0) = J^S(0) \qquad \text{and} \qquad \mathbf{J}^S(i) = \mathbb{E}_r[J^S(i,r)] = \int_{r \in \Re+} J^S(i,r) f(r) dr, \quad \forall i \geq 1. \tag{3.20}$$

We can re-write the Bellman equation for the problem with strategic agents as follows:

$$J^S(i,r) = \min_{\phi \in \Theta} \left((1 - w(\phi,r)) \mathbf{J}^S(i-1) + w(\phi,r) J^S(i,r,\phi) \right) \qquad \forall i \geq 1 \quad r \in \Re+, \tag{3.21}$$

$$J^S(i,r,\phi) = \frac{1}{\beta + \lambda + \bar{\mu}} \left(g(i) + \lambda J^S(i+1,r,\phi) + (\bar{\mu} - \mu_\phi) J^S(i,r,\phi) + \mu_\phi \mathbf{J}^S(i-1) + \mu_\phi R^S_\phi(r) \right)$$

$$\forall i \geq 1 \quad r \in \Re + \quad \phi \in \Theta, \tag{3.22}$$

$$J^S(0) = \frac{1}{\beta + \bar{\mu} + \lambda} \left(g(0) + \lambda \mathbf{J}^S(1) + \bar{\mu} J^S(0) \right). \tag{3.23}$$

The first term of Equation (3.21) captures the event where the agent reneges, which occurs with probability $1 - w(\phi, r)$. The second term captures the event where the agent goes through screening; in this case, the system transition instantaneously to state (i, r, ϕ). The associated cost then includes the continuous queuing cost $g(i)$ and the expected risk cost $R^S_\phi(r)$ if the next event is a service completion. Upon transition, it will then incur a cost-to-go equal to $J^S(i+1, r, \phi)$ if the next event is an arrival, a cost-to-go equal to to $J^S(i, r, \phi)$ if the next event is a self-transition, and an expected cost-to-go equal to $\mathbf{J}^S(i-1)$ if the next event is a service completion. This is captured in Equation (3.22). Equation (3.23) treats the case where $i = 0$.

Note that the problem's formulation proposed here captures in a single Bellman equation the endogeneity of agents' strategic behaviors in response to the selection of screening procedures by the operator. Unfortunately, the dynamics of the system become much more complex, which hinders the problem's analytical tractability. Nevertheless, in the next section, we derive some properties of the dynamics of the system and supplement these with numerical results assessing system performance in the presence of strategic agents.

3.6 Impact of Strategic Behavior on System Performance:

We now discuss the impact of strategic behaviors on the optimal policy and resulting system performance. As we shall see, strategic behaviors lead to stricter screening, to force malevolent agents out of the system. By applying slower screening procedures, the operator can improve system performance by reducing risk costs, and even by reducing expected queue lengths when the proportion of malevolent agents is large enough (Section 3.6.1). We then analyze the impact of pre-screening information. We find that more reliable pre-screening information leads to lower expected costs, but not necessarily to lower risk costs (Section 3.6.2).

To isolate the effects of strategic behaviors and pre-screening information, we consider a setting with two agent types, with true risk profiles $s_1 \leq s_2$—i.e., agents of type s_1 are benevolent and agents of type s_2 are malevolent. We also consider two risk levels $r_1 \leq r_2$. The mapping is shown in Figure 3.4: Any agent in category r_1 has a probability α_1 (resp. $1 - \alpha_1$) of being of type s_1 (resp. s_2) and any agent in category r_2 has a probability α_2 (resp. $1 - \alpha_2$) of being of type s_2 (resp. s_1). We denote by r (resp. $1 - r$) the probability that an agent has a risk level r_1 (resp. r_2). To satisfy Equation (3.13), we assume that the probability an r_2 agent is malevolent is greater than that an r_1 agent is, or $1 - \alpha_1 \leq \alpha_2$, i.e., $\alpha_1 + \alpha_2 \geq 1$.

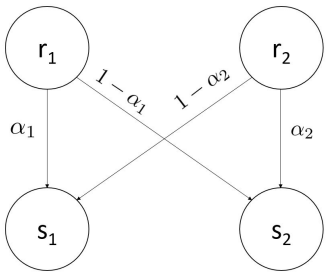


Figure 3.4: Mapping between true profiles (s_1, s_2) and risk levels (r_1, r_2).

Note that we focus here on a different aspect of pre-screening information from Section 3.4. Indeed, Section 3.4 focused on the benefits of greater *extents* of pre-screening information by comparing the multi-class policy to a baseline single-class policy. In contrast, we now fix the number of job classes but analyze the benefits of higher *quality* pre-screening information—the parameters α_1 and α_2 control the information asymmetries between the agents and the screening operators.

On the service side, we consider two screening procedures: "slow" vs. "fast", represented by subscripts S and F, respectively. We assume that, under fast service, all agents remain in the system; in contrast, slow service forces malevolent agents to renege from the system. Mathematically, we have:

$$p_F > \tau^{s_1} > p_S > \tau^{s_2}, \tag{3.24}$$

$$S_S^+ = \{s_1\} \quad \text{and} \quad S_F^+ = \{s_1, s_2\}. \tag{3.25}$$

Our insights hold in more general instances, but we focus on this setting for expositional ease.

3.6.1 Impact of Strategic Behaviors

We analyze the impact of strategic behaviors on system performance. First, we analytically compare the impact of two policies—one more conservative than the other—on the dynamics of the system and the resulting risk and queuing costs. We show that, in the presence of strategic agents, stricter (more conservative) screening may yield a Pareto improvement, i.e., lower risk costs as well as lower expected queue lengths. This is unusual, as slower procedures typically result in higher congestion but, in our setting, this can be outweighed by higher reneging incidence. We then show computationally that the optimal policy with strategic behaviors is more conservative than the one without strategic behaviors; in other words, reneging induces stricter screening to force malevolent agents our of the system. Under some conditions, this stricter screening policy results in lower expected queue lengths than the faster screening policy applied when strategic behaviors are not considered—confirming that the analytical insights hold under the optimal policy.

Analytical Insights:

We consider two specific policies; they may not be optimal, but they show that, under some conditions, applying slower screening procedures in the presence of strategic agents may result in lower risk costs as well as lower expected queue lengths. Specifically, we consider: (i) an "expedited" policy, under which the fast screening procedure is always applied, and (ii) a more "conservative" policy, under which the fast screening procedure is applied unless if there is exactly one agent with risk level r_2 in the system. We refer to these two policies with superscript e and c, respectively. Formally, we have:

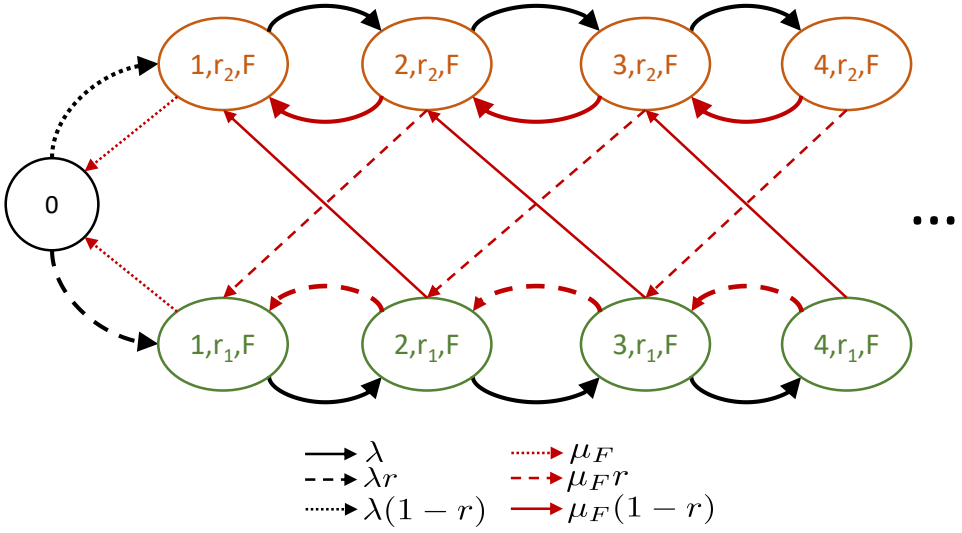
$$\pi^e(i, r) = F, \quad \forall i \geq 1, r \in \{r_1, r_2\}. \tag{3.26}$$

$$\pi^c(i, r) = F, \quad \forall i \geq 2, r \in \{r_1, r_2\}, \quad \pi^c(1, r_1) = F, \quad \text{and } \pi^c(1, r_2) = S. \tag{3.27}$$

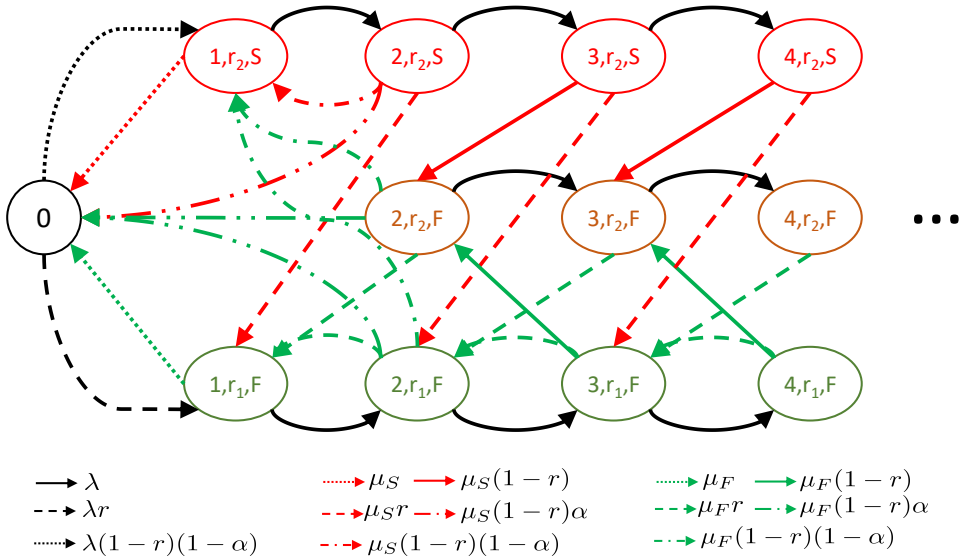
We also assume that $\alpha_1 = 1$ and $\alpha_2 = \alpha$. In other words, each agent with risk level r_1 is necessarily of type s_1 (i.e., a benevolent agent), and each agent with risk level r_2 may be of type s_2 (i.e., a malevolent agent) with probability α or of type s_1 (i.e., a benevolent agent) with probability $1 - \alpha$.

The Markov chain characterizing the system dynamics under each policy is depicted in Figure 3.5. Under the expedited policy (Figure 3.5a), no reneging occurs as the fast screening is applied regardless of the state

of the system. In contrast, under the conservative policy (Figure 3.5b), a malevolent agent reneges if she is screened while no one is in queue. From the perspective of the operator, applying the slow screening in state $(1, r_2, S)$ results in reneging with probability α. For instance, in states $(2, r_1, F)$ and $(2, r_2, F)$, upon each service completion (occurring at rate μ_F), the system transitions to state $(1, r_1, F)$ with probability r (the next agent is of risk level r_1), to state $(1, r_2, S)$ with probability $(1-r)(1-\alpha)$ (the next agent is of risk level r_2 and of type s_1), and to state 0 with probability $(1-r)\alpha$ (the next agent is of risk level r_2 and of type s_2).



(a) Expedited policy



(b) Conservative policy

Figure 3.5: Markov chain under the expedited and conservative policies.

We denote by $\nu^e(i, r, \phi)$ (resp. $\nu^c(i, r, \phi)$) the steady-state probability of state (i, r, ϕ) and by $\nu^e(0)$ (resp. $\nu^c(0)$) the steady-state probability of state 0 under the expedited (resp. conservative) policy. We denote

by $E[N^e]$ and $E[N^c]$ the expected queue lengths and by $\xi^e(r)$ and $\xi^c(r)$ the expected risk cost induced by screening an agent of risk level r, under the expedited and conservative policies, respectively. The expected risk cost is equal to the expectation of $R^S_\phi(r)$ given the agent's risk level r, conditionally on the operator being busy. Mathematically, these performance metrics are given by:

$$E[N^j] = \sum_{i\geq 0} \sum_{r\in\{r_1,r_2\}} \sum_{\phi\in\{S,F\}} i \times \nu^j(i,r,\phi) \qquad \forall j \in \{e,c\}. \tag{3.28}$$

$$\xi^j(r) = \frac{\sum_{\phi\in\{S,F\}} \sum_{i\geq 1} \nu^j(i,r,\phi) \times R^S_\phi(r)}{\sum_{\phi\in\{S,F\}} \sum_{i\geq 1} \nu^j(i,r,\phi)} \qquad \forall j \in \{e,c\}. \tag{3.29}$$

Proposition 3.4 (proved in Appendix C.3) compares the impact of the expedited policy and the conservative policy on system performance, in the presence of strategic agents.

Proposition 3.4. *Under policies $\pi^e(i,r)$ and $\pi^c(i,r)$, the following results hold:*

$$\xi^e(r_1) \geq \xi^c(r_1) \text{ and } \xi^e(r_2) \geq \xi^c(r_2). \tag{3.30}$$

$$E^c[N] \leq E^e[N] \text{ if and only if } \alpha \geq \frac{(\lambda+\mu_S)(\mu_F-\mu_S)}{\lambda(\mu_F-\mu_S)+\mu_F\mu_S}. \tag{3.31}$$

First, the conservative policy leads to a lower expected risk cost for both agent types than the expedited policy. This is intuitive, as slower screening leads to a smaller cost of false negatives as well as a greater incidence of reneging. More surprisingly, the conservative policy may *also* lead to lower expected queue lengths when α exceeds a threshold given by $\frac{(\lambda+\mu_S)(\mu_F-\mu_S)}{\lambda(\mu_F-\mu_S)+\mu_F\mu_S}$. In other words, stricter screening might lead to a Pareto improvement. This stems from the fact that, for higher values of α, more malevolent agents may renege under the conservative policy; the resulting reduction in the number of agents going through screening can offset the negative impact of slower screening on queue lengths. In contrast, for lower values of α, the expected queue lengths are lower under the expedited policy—i.e., the increased reneging is not sufficient to outweigh the queue length increase resulting from slower screening.

We note, the threshold $\frac{(\lambda+\mu_S)(\mu_F-\mu_S)}{\lambda(\mu_F-\mu_S)+\mu_F\mu_S}$ decreases with μ_S, increases with μ_F, and increases with λ. First, as the slow screening procedure gets faster, the conservative policy gives up less when utilizing slow service and thus is more likely to lead to smaller expected queue lengths than the expedited one. Conversely, as the fast screening option gets faster, the negative effect on queue lengths of the conservative policy becomes relatively stronger, and the conservative policy is thus less likely to reduce expected queue lengths. Finally, as demand increases, higher probability mass is assigned to states with $i \geq 2$ jobs in the system; as a result, the incidence of reneging diminishes, and the conservative policy is less likely to reduce expected queue lengths.

In conclusion, slower screening policies can lead to lower risk costs as well as lower expected queue lengths, due to the opposite effects of slower service vs. higher incidence of reneging. Next, we identify the impact of agents' strategic reneging on the operator's *optimal* policies and resulting system performance.

Computational Results

We identify the optimal policy with reneging behavior (referred to as "*Policy R*"), and compare it to the baseline policy obtained without the reneging behavior (referred to as "*Policy NR*"). We initially consider the same mapping as earlier, i.e., $\alpha_1 = 1$ and $\alpha_2 = \alpha \in [0,1]$.

Note that we have proved formally that *Policy NR* exhibits a threshold behavior (Section 3.3) but, due to the complexity of the system dynamics with strategic agents, we have not shown that *Policy R* also exhibits a threshold behavior. Nonetheless, this is verified numerically across a range of parameter values, as long as information on risk levels is not extremely inaccurate (Assumption 6).

Table 3.1 characterizes the two policies by showing their thresholds as a function of α and model parameters (a "Base" setting, settings with higher risk and queuing costs, and settings with more or less differentiated screening speeds). For instance, in the Base setting with $\alpha = 0.3$, *Policy NR* applies the "slow" procedure when $i < 2$ and the "fast" procedure when $i \geq 2$ for agents of risk level r_2; it applies the "fast" procedure for agents of risk level r_1 regardless of the queue length. When no threshold is reported, then the "slow" policy is applied for all values of $i \geq 1$.

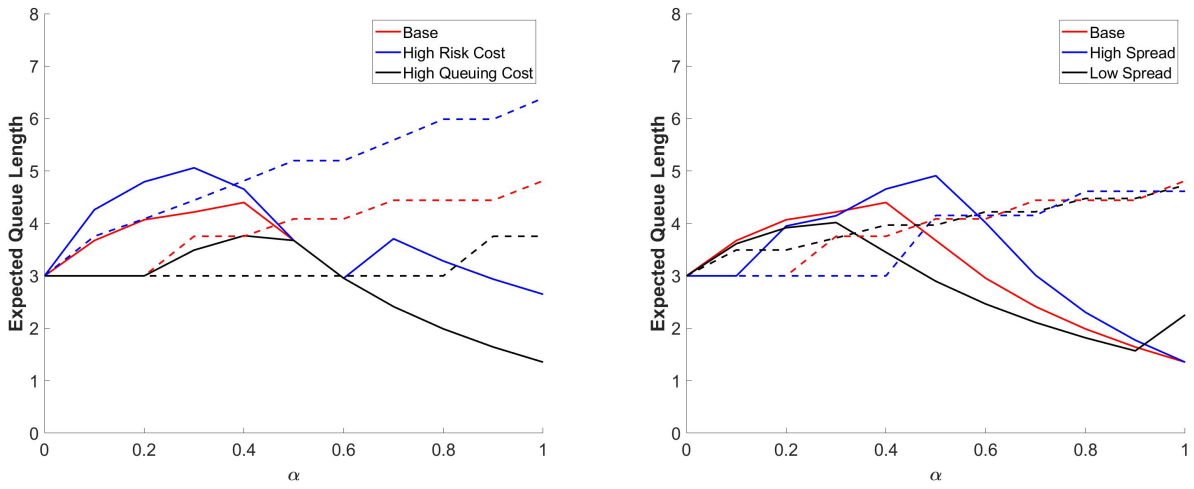
Table 3.1 shows that *Policy NR* has a consistently lower threshold than *Policy R*. This suggests that, for all parameter values, accounting for agents' strategic behaviors induces slower (i.e., more conservative) optimal policies than those obtained in the absence of strategic behaviors. In other words, *Policy R* deliberately deters malevolent agent from going through screening by applying stricter screening.

Table 3.1: Thresholds of *Policy R* and *Policy NR*, above which fast service is applied.

	Base				High Risk				High Queuing				High Spread				Low Spread			
	r_1		r_2		r_1		r_2		r_1		r_2		r_1		r_2		r_1		r_2	
α	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*
0.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.1	1	1	1	2	1	1	2	4	1	1	1	1	1	1	1	1	1	1	2	3
0.2	1	1	1	4	1	1	3	7	1	1	1	1	1	1	1	2	1	1	2	6
0.3	1	1	2	6	1	1	4	12	1	1	1	2	1	1	1	3	1	1	3	14
0.4	1	1	2	14	1	1	5	28	1	1	1	5	1	1	1	6	1	1	4	85
0.5	1	1	3	85	1	1	6	86	1	1	1	83	1	1	2	13	1	1	4	92
0.6	1	1	3	91	1	1	6	91	1	1	1	90	1	1	2	84	1	1	5	-
0.7	1	1	4	-	1	2	7	-	1	1	1	-	1	1	2	89	1	1	5	-
0.8	1	1	4	-	1	2	8	-	1	1	1	-	1	1	3	-	1	1	6	-
0.9	1	1	4	-	1	2	8	-	1	1	2	-	1	1	3	-	1	1	6	-
1.0	1	1	5	-	1	2	9	-	1	1	2	-	1	1	3	-	1	2	7	-

Next, Figure 3.6 shows the expected queue length as a function of α in the same settings as Table 3.1; *Policy NR* and *Policy R* are depicted with dashed and straight lines, respectively. Note that *Policy R* leads to lower expected queue lengths than *Policy NR* for the larger values of α. This is consistent with the insights derived in Proposition 3.4, but extends them to the optimal policies: When the effect of reneging behaviors is large enough (i.e. for the larger values of α), then a slower screening policy (i.e., *Policy R*) may lead to lower expected queue lengths than a faster screening policy (i.e., *Policy NR*) and thus Pareto improvement under strategic behavior, by reducing both risk levels and expected queue lengths. In contrast, for lower values of α, the effect of reneging is not sufficient to compensate the slower screening speeds, and *Policy R* results in longer expected queues than *Policy NR*. Moreover, note from Figure 3.6b that *Policy R* is more likely

to lower expected queue lengths (in addition to risk levels) as the service rates become less differentiated —which is, again, consistent with the analytical results.



(a) Variations in risk and queuing costs

(b) Variations in spreads in service speeds

Figure 3.6: Expected queue lengths as a function of α [*Policy NR* in dashed lines, *Policy R* in solid lines].

Finally, we extend these findings to a more general mapping function by varying the values of α_1 and α_2 jointly (instead of fixing $\alpha_1 = 1$). Results are shown in Table 3.2 and Figure 3.4. As in Table 3.1, *Policy R* has higher threshold levels than *Policy NR*, confirming that incorporating agents' strategic behaviors results in slower screening. As α_1 decreases, *Policy R* becomes more conservative for agents with risk level r_1, reflected by the higher thresholds in Table 3.2—since a higher fraction of agents of risk level r_1 are malevolent—resulting in higher expected queue lengths. Nevertheless, *Policy R* (i.e., stricter screening) may still lead to lower expected queue lengths than *Policy NR* but at higher values of α_2, than the $\alpha_2 = 1$ case.

Summary

Agents' strategic behaviors have a significant impact on optimal screening policies and resulting system performance. First, strategic agent behaviors lead to stricter (more conservative) screening than non-strategic agents. This mitigates expected risk costs by deterring some malevolent agents and reducing the probability of false negatives for each screening. However, the effect on queue lengths is unclear. On the one hand, slower screening has a negative impact on queue lengths but, at the same time, more conservative screening can reduce queue lengths by inducing a higher incidence of reneging among malevolent agents. As our results show, the net impact of more conservative policies on queue lengths may actually be positive. In other words, stricter, slower screening procedures may achieve a Pareto improvement in the presence of strategic agents by simultaneously reducing the expected risk costs and the expected queue lengths. Importantly, stricter screening is more likely to reduce expected queue lengths when reneging behaviors are stronger (higher values of α_2) and information asymmetries are weaker (higher values of α_1).

From the system's perspective, appropriate screening policies depend on the underlying risk profiles of the population and the information available to the system operator. For instance, at border crossings with

	$\alpha_1 = 0.65$				$\alpha_1 = 0.8$				$\alpha_1 = 1$			
	r_1		r_2		r_1		r_2		r_1		r_2	
α_2	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*	*NR*	*R*
0.50	4	17	6	86	3	6	6	86	1	1	5	86
0.55	4	17	7	88	3	6	7	88	1	1	6	89
0.60	4	17	7	91	3	6	7	91	1	1	6	91
0.65	4	17	8	-	3	6	7	-	1	1	7	-
0.70	4	17	8	-	3	7	8	-	1	1	7	-
0.75	4	17	9	-	3	7	8	-	1	1	7	-
0.80	4	17	9	-	3	7	9	-	1	1	8	-
0.85	4	17	10	-	3	7	9	-	1	1	8	-
0.90	4	18	10	-	3	7	9	-	1	1	8	-
0.95	4	18	11	-	3	7	10	-	1	1	9	-
1.00	4	18	11	-	3	7	10	-	1	1	9	-

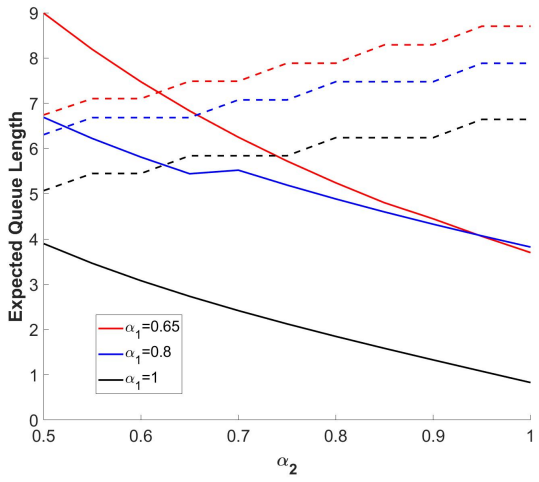


Figure 3.7 & Table 3.2: Thresholds of *Policy R* and *Policy NR*, and expected queue lengths.

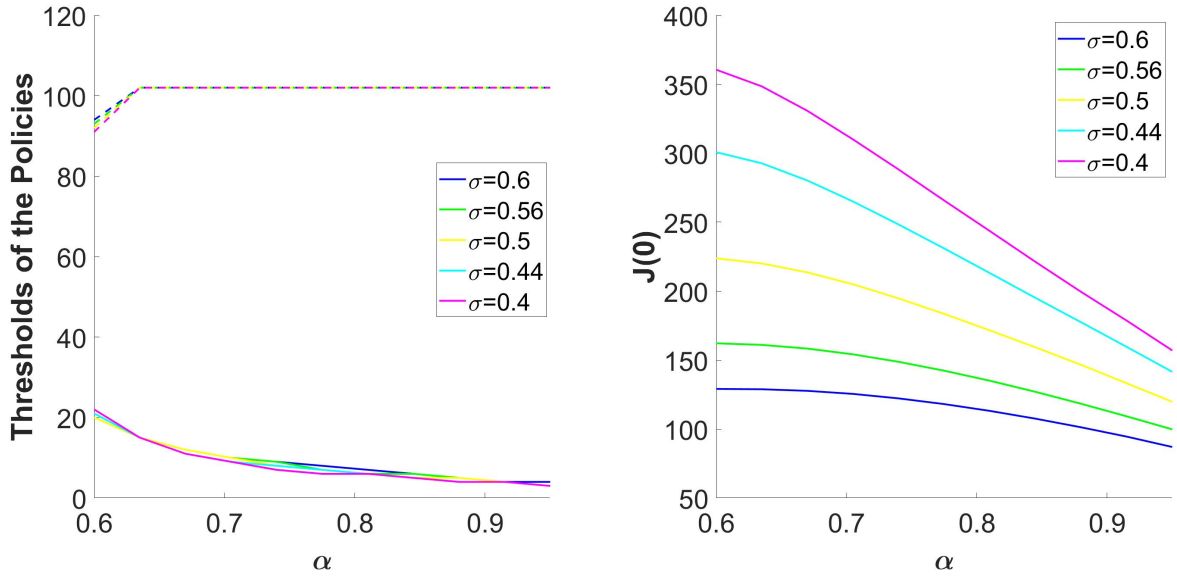
a significant proportion of travelers carrying potentially illicit items (from minor custom law violations to major ones), strict screening may deter a number of malevolent behaviors—either by inducing reneging, as modeled here, or by deterring people from joining the system in the first place. Thus, conservative screening may lower expected risk costs as well as congestion levels. At the other extreme, airport security operations typically face a much lower proportion of malevolent agents. As a result, more conservative policies can mitigate risk by preventing false negatives, but typically also increase system congestion.

3.6.2 Impact of Pre-screening Information

We conclude this section by examining the value of pre-screening information and its effect on system performance. We model the quality of pre-screening information by varying information asymmetries between the agents and the operator. Specifically, we fix the proportion of agents of type s_2 in the population, and denote it by σ. We then parametrize the mapping function by setting $\alpha_1 = \alpha_2 = \alpha$, so α measures the accuracy of the profiling information—with $\alpha > 0.5$ to satisfy Equation (6). The distribution of risk profiles is then determined by the parameter $r = \frac{\alpha-\sigma}{2\alpha-1} \in [0, 1]$, which is the probability a customer has profile r_1.

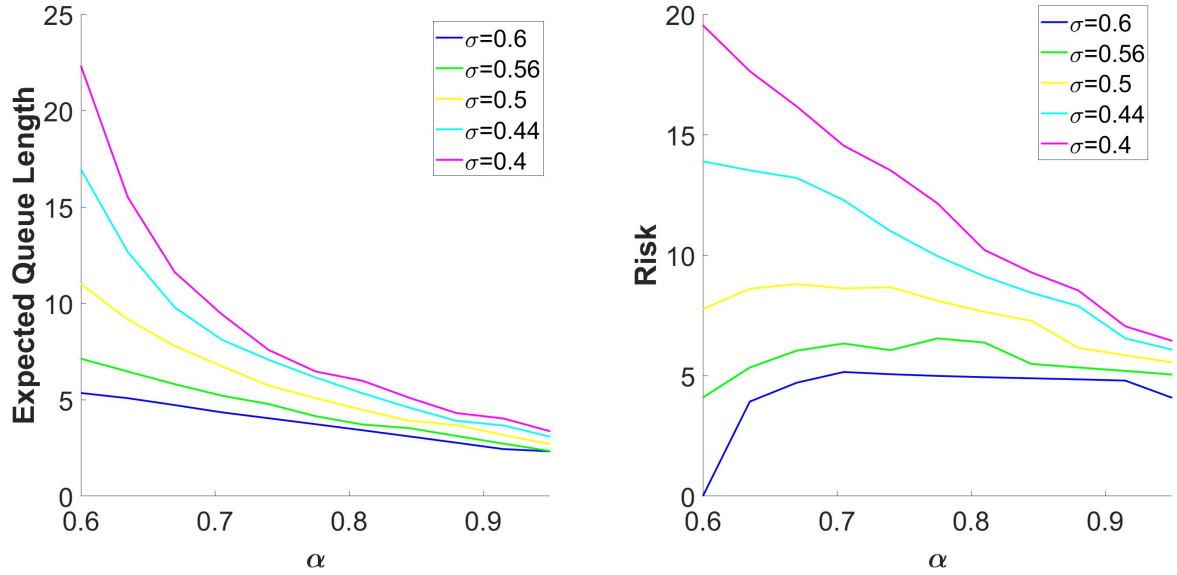
Figure 3.8 shows the optimal policy, the total discounted cost, the expected queue lengths, and the risk levels as a function of α, for various values of σ, the probability that an agent is malevolent, Figure 3.8a plots the threshold for fast service of the optimal policy for agents of risk level r_1 and r_2 with straight and dashed lines, respectively. The total discounted cost is shown as $J(0)$, as in Section 3.4.2. The expected risk and expected queue length are computed from Equations (3.28) and (3.29).

First, note from Figure 3.8a that, as α increases, the optimal screening policy becomes looser for agents with risk level r_1 and stricter for agents with risk level r_2. In other words, better pre-screening information, i.e., lower information asymmetries, result in better screening discrimination across risk categories. Second, as expected, better profiling information leads to lower total costs, as shown in Figure 3.8b. Note also that total costs are *decreasing* in σ, due to the system's ability to induce higher reneging for larger proportions of malevolent agents of type s_2. Next, better profiling information leads to lower expected queue lengths



(a) Threshold Levels

(b) Total Costts



(c) Queue Length

(d) Risk Levels

Figure 3.8: Optimal policy and metrics for different service speeds and risk profiles.

(Figure 3.8c). This stems from the facts that looser screening for agents with risk level r_1 directly increases service speeds, and that stricter screening for agents with risk level r_2 increases the incidence of reneging.

More surprisingly, Figure 3.8d shows that the expected risk cost does not vary monotonically with α, i.e., risk costs do *not* necessarily decrease as information asymmetries get lower. This stems from two main factors. On the one hand, better information enables the operator to discriminate across the two risk profiles, thus applying stricter screenings for agents with risk level r_2 and reducing the corresponding risk. At the

same time, this discrimination results in looser screening for agents with risk level r_1. When information asymmetries are fairly strong (i.e., for lower values of α) and the underlying population exhibits high risks (i.e., for high values of σ), a high proportion of malevolent agents of type s_2 are in risk category r_1. As a result, looser screenings can induce a high risk cost—which can, in fact, dominate the risk mitigation resulting from applying stricter screening to agents of type r_2. In this case, the system is willing to tolerate increased risks in order to accelerate service of mostly benevolent r_1 agents. In contrast, when information asymmetries are lower and the overall population is less risky, enhanced information achieves a Pareto improvement by mitigating both risk and queuing costs because malevolent agents are more concentrated in risk profile r_2.

In summary, enhanced pre-screening information enables the operator to apply differentiated screening procedures across the various risk categories—e.g., looser screening for low-risk agents and stricter screening for high-risk agents. This reduces expected queue lengths by speeding up screening for low-risk agents and inducing more reneging from high-risk agents. However, it does not necessarily result in lower risk costs—especially when information asymmetries are strong and the underlying population induces high risks.

3.7 Conclusion

This chapter has proposed an original approach to balance *efficiency* and *risk* in security screening operations. On the one hand, faster service can mitigate wait times but creates higher risks of false negatives. Vice versa, more conservative screening can mitigate security risks but come at the expense of slower service. We have formulated Markov decision processes to dynamically select screening procedures as a function of observed queue lengths and the risk level of each job. This chapter extends the literature on speed-quality trade-offs in two ways. First, it considers a multi-class setting with heterogeneous jobs, leveraging pre-screening information on heterogeneous risk levels. Second, it captures strategic behaviors of the agents, who may elect to renege prior to screening if the risk of being caught is too high.

The main results from this chapter fall into four main categories. First, the optimal policy exhibits a double threshold behavior: All else equal, stricter screening is applied when the queue is shorter and/or when the job is associated with a higher risk level. Second, leveraging job-level risk information can reduce expected costs by up to 6-7%, compared to baseline single-class decision-making schemes based on queue lengths alone. Third, anticipating agents' strategic behaviors results in stricter screenings. These slower procedures lower expected risks but may also, surprisingly, lower expected queue lengths by forcing malevolent agents out of the system. Fourth, reductions in information asymmetries between agents and operators through better pre-screening information lowers expected costs; however, this may come with increased expected risks.

From a practical standpoint, this chapter provides decision-making tools to support screening operations at major security checkpoints (e.g., seaports, airports, land borders, military facilities, etc.). At the tactical level, the proposed models can be used to select differentiated screening procedures for heterogeneous agents. Our results also motivate the use of stricter screening procedures in settings with human agents (e.g., border crossings) than in those without (e.g., package and container screening). At the strategic level, our results stress the benefits of designing a menu of screening procedures, and granting flexibility to screening operators to select the appropriate procedure as a function of observed queue lengths and risk information. Finally, this chapter highlights two operational benefits of pre-screening risk information: (i) characterizing agent heterogeneity by creating multi-class job profiles, and (ii) reducing information asymmetries between agents and system operators.

Our results also motivate further research on risk-efficiency trade-offs in screening systems. First, the model developed in this chapter involves a number of assumptions and simplifications; in future work, these could be relaxed by considering additional complexities arising in practice, such as systems with multiple screening operators, priorities among jobs, dedicated lanes for some jobs, sequential screenings, etc. Second, the policies considered here could be enhanced with probabilistic selections of screening procedures. Third, this chapter has assumed simplified functional forms to characterize agent types and system-wide risk costs; such assumptions should be validated empirically. Finally, an important question for empirical researchers lies in the characterization of screening procedures applied in practice by operators, and how they compare with the analytical predictions developed in this chapter. This study lays the foundations to tackle these questions, potentially facilitating more efficient and robust security screening systems.

Chapter 4

Conclusion

Timeliness is a key value proposition of operations. In this dissertation we have examined three problems that require timely delivery of service operations. Using game theory, queuing theory, machine learning and optimization we propose policies and solution methodologies with the goal of real-life implementations.

In Chapter 1, we propose a solution to the problem of ambulance dispatch in Emergency Medical Services with a novel data-driven methodology that combines ideas from machine learning and decision making under uncertainty. Our method extends the operations literature on dispatching of emergency medical services and sets a new benchmark for dispatching problem. In collaboration with University of Pittsburgh Medical Center and Allegheny County Emergency Medical Services we test our methodology on unique data set from city of Pittsburgh, Pennsylvania.

As part of the solution methodology, we first develop a omniscient deterministic binary linear program that incorporates forward looking dispatch decisions. Using the deterministic model as a basis we build a scenario based robust optimization formulation that produces dispatch decisions under future uncertainty. Scenario based robust optimization is a novel modelling approach that combines ideas from stochastic programming and robust optimization by probabilistic scenario sets and protecting against adversarial scenarios within the scenario sets. To this end, we also develop the Closest Neighbours Clustering method, a data-driven methodology for building the scenario sets from real data. We show the benefits of using scenario based robust optimization along with Closest Neighbours Clustering against a series of benchmarks including stochastic programming and robust optimization. Moreover, our methodology outperforms the state-of-the art dispatching methods in the existing literature by a significant margin.

In Chapter 2, we study an on-demand platform's optimal delay information disclosure when the platform matches two classes of users (consumers and providers) with the objective of maximizing the effective match rate (as a proxy for revenue). We study three information regimes— *occupancy*, where both consumers and providers are informed of the current system occupancy, and two asymmetric information regimes, where either only consumers or only providers are informed of the current system occupancy, while the other is provided with no information.

Our base model considers users of each class as being either patient or impatient; we analytically compute and compare the platform's match rates under the three information regimes and provide sufficient conditions for each regime's optimality. We further find that it is optimal for platform to share no delay information with its users under certain limiting settings. Through numerical analysis, we show that user's welfare is not

always aligned with platform's profitability and only aligned under occupancy information regime. We extend our base model by studying how the platform's information regime choice changes when user delay sensitivity has a higher heterogeneity than in the two-point distribution. As users become more heterogeneous in their patience, the platform finds it optimal to disclose occupancy information for a larger region of the parameter space.

In Chapter 3, we propose an original approach to balance *efficiency* and *risk* in security screening operations. On the one hand, faster service can mitigate wait times but creates higher risks of false negatives. Vice versa, more conservative screening can mitigate security risks but come at the expense of slower service. We have formulated Markov decision processes to dynamically select screening procedures as a function of observed queue lengths and the risk level of each job. This chapter extends the literature on speed-quality trade-offs in two ways. First, it considers a multi-class setting with heterogeneous jobs, leveraging pre-screening information on heterogeneous risk levels. Second, it captures strategic behaviors of the agents, who may elect to renege prior to screening if the risk of being caught is too high.

The main results from this chapter fall into four main categories. First, the optimal policy exhibits a double threshold behavior: All else equal, stricter screening is applied when the queue is shorter and/or when the job is associated with a higher risk level. Second, leveraging job-level risk information can reduce expected costs by up to 6-7%, compared to baseline single-class decision-making schemes based on queue lengths alone. Third, anticipating agents' strategic behaviors results in stricter screenings. These slower procedures lower expected risks but may also, surprisingly, lower expected queue lengths by forcing malevolent agents out of the system. Fourth, reductions in information asymmetries between agents and operators through better pre-screening information lowers expected costs; however, this may come with increased expected risks.

This dissertation also opens a series of new questions to the researchers for the problems we have examined. Related to Chapter 1, the dispatching problem of Emergency Medical Services, different urgency levels for emergency calls as well as different types of ambulances that provide different levels of care are two features we have not included in our modeling. Moreover, implementation of Closest Neighbour Clustering and scenario based robust optimization for other data-sets could be an interesting research avenue for showing benefits of these methodologies. Related to Chapter 2, we have looked at the problem of information delay in a setting with linear cost structure for user's utility. It would be an interesting to study the effect of users having non-linear delay cost functions. While we study only three practical information regimes, it would be useful to expand the space of regimes to find the structure of the "optimal" information regime. Finally, another interesting extension would be to study service disciplines other than first-come-first-served (for example, a random service discipline), to better suit particular applications (for example, ride sharing). Lastly, related to Chapter 3, the model developed in this chapter involves a number of assumptions and simplifications; in future work, these could be relaxed by considering additional complexities arising in practice, such as systems with multiple screening operators, priorities among jobs, dedicated lanes for some jobs, sequential screenings, etc. Moreover, the policies considered here could be enhanced with probabilistic selections of screening procedures.

Appendices

Appendix A

Appendices for Chapter 1

A.1 Proofs of Propositions

A.1.1 Proof of Proposition 1.1

There are 4 properties we show for $D^s(i,j)$:

- Non-negativity i.e., $D^s(\cdot,\cdot) > 0$.
- Symmetry, i.e., $D^s(i,j)) = D^s(j,i)$.
- Identity of indiscernibles i.e., $D^s(i,j) = 0 \iff i = j$.
- Triangle inequality, $D^s(i,k) \leq D^s(i,j) + D^s(j,k)$

The first two properties are trivial and they are direct result of definition of D^s and the fact that D^s is driven based on between-calls-metric d^c.

We prove the third and fourth properties below. Without loss of generality we assume that $k \leq j \leq i$. Hence we can write the distances of three scenarios as follows:

$$D^s(i,j) = \max_{c^m \in s_j(c)} \big(\min_{c^n \in s_i(c)} d^c(c^m, c^n) \big)$$
$$D^s(i,k) = \max_{c^m \in s_k(c)} \big(\min_{c^n \in s_i(c)} d^c(c^m, c^n) \big)$$
$$D^s(j,k) = \max_{c^m \in s_k(c)} \big(\min_{c^n \in s_j(c)} d^c(c^m, c^n) \big)$$

We start with the third property i.e., $D^s(i,j) = 0 \implies i = j$. First, assume that $D^s(i,j) = 0$, then we have that $\min_{c^n \in s_i(c)} d^c(c^m, c^n) = 0 \ \forall c^m \in s_j(c)$. Then $\arg\min_{c^n \in s_i(c)} d^c(c^m, c^n) = c_i^j \forall c^m \in s_j(c)$. Since d^c is a proper metric we have that $d^c(c^m, c_i^j) = 0 \iff c^m = c_i^j$. Then for every $c^m \in s_j(c)$ there is a call $c_i^j \in s_i(c)$ such that $c^m = c_i^j$. Since we also know that $j \leq i$, we conclude that $j = i$

Next assume that $s_i(c) = s_j(c)$. Then for each $c^m \in s_j(c)$, we have that $\min_{c^n \in s_i(c)} d^c(c^n, c^m) = 0$. Then taking the $\max_{c^m \in s_j(c)} \min_{c^n \in s_i(c)} d^c(c^m, c^n) = 0$. Hence, we are done for the third property.

Next we show the fourth and the last property which is the triangle inequality:

$$D^s(i,k) \leq D^s(i,j) + D^s(j,k) \tag{A.1}$$

Let us first define the following calls:

- $c_*^k = \arg\max_{c^m \in s_k(c)} \left(\min_{c^n \in s_i(c)} d^c(c^m, c^n) \right)$.
- $\bar{c}^j = \arg\min_{c^n \in s_j(c)} d^c(c^n, c_*^k)$.
- $\bar{c}^i = \arg\min_{c^n \in s_i(c)} d^c(c^n, \bar{c}^j)$.

Next, observe the followings inequalities:

$$d^c(c_*^k, \bar{c}^i) \geq D^s(i,k), \text{ due to inner minimization.} \tag{A.2}$$
$$d^c(c_*^k, \bar{c}^j) \leq D^s(k,j), \text{ due to outer maximization.} \tag{A.3}$$
$$d^c(\bar{c}^i, \bar{c}^j) \leq D^s(i,j), \text{ due to outer maximization.} \tag{A.4}$$
$$d^c(c_*^k, \bar{c}^i) \leq d^c(c_*, \bar{c}^j) + d^c(\bar{c}^j, \bar{c}^i) \text{ due to triangular inequality of } d^c(i,j). \tag{A.5}$$

Then, we combine these inequalities as follows:

Using (A.2) and (A.5) we obtain:

$$D^s(i,k) \overbrace{\leq}^{\text{due to (A.2)}} d^c(c_*^k, \bar{c}^i) \overbrace{\leq}^{\text{due to (A.5)}} d^c(c_*^k, \bar{c}^j) + d^c(\bar{c}^j, \bar{c}^i)$$

Then using (A.3) and (A.4) we obtain:

$$D^s(i,k) \leq \underbrace{d^c(c_*^k, \bar{c}^j) + d^c(\bar{c}^j, \bar{c}^i) \leq D^s(i,j) + D^s(j,k)}_{\text{due to (A.3) and (A.4)}}$$

Hence we are done.

A.1.2 Proof of Proposition 1.2:

For reference, we first present the offline BLP of Jagtenberg et al. (2017b):

$$\min_{x_{cv}} \sum_{c \in \mathcal{S}} \sum_{v \in B} l_{cv} x_{cv} \tag{A.6}$$
$$s.t. \sum_{v \in B} x_{cv} = 1, \qquad c \in \mathcal{S} \tag{A.7}$$
$$o_{cc'v}(x_{cv} + x_{c'v}) \leq 1, \qquad v \in B,\ c, c^{'} \in \mathcal{S}\ c \neq c^{'} \tag{A.8}$$
$$x_{cv} \in \{0,1\} \tag{A.9}$$

In this formulation, similar to term a_{cc^2v} , $o_{cc'v}$ is an indicator term that takes value 1 if emergency calls' c and $c^{'}$ have overlapping time frames when responded by ambulance v. Jagtenberg et al. (2017b) assume that if there is an overlap, $o_{cc'v} = 1$, than vehicle v can respond to only one of two emergency calls c and $c^{'}$.

Let $\mathbf{x}^{BLP} := \{x_{cv} | c \in \mathcal{S}, v \in \mathcal{B}\} \in X^{BLP}$ be a feasible solution in the offline BLP of Jagtenberg et al. (2017b). We show that $\mathbf{x}^{BLP} \in X^{D-BLP}$. We will do so by constructing an equivalent feasible solution in our model. We construct $\mathbf{x}^{D-BLP} := \{x_{cc^2v} | c \in \mathcal{S}, c \in \mathcal{C}^2(c), v \in \mathcal{B}\} \in X^{D-BLP}$ such that $x_{cAv} = x_{cv}$ $\forall c \in \mathcal{S}, v \in \mathcal{B}$ and $x_{cc^2v} = 0$ $\forall c \in \mathcal{S}, v \in \mathcal{B}, c^2 \in \mathcal{C}(c)/\{A\}$.

Note that (A.7) ensures that every call $c \in \mathcal{S}$ is responded to by 1 ambulance and accordingly, the feasible solution $\mathbf{x}^{D-BLP}$ sends an ambulance to every call $c \in \mathcal{S}$ from the availability state. This satisfies constraint (1.2). By construction of $\mathbf{x}^{D-BLP}$ and the definition of term a_{cc^2v} (in which $a_{cAv} = 1$) (1.4) is trivially satisfied. In similar spirit, due to implications of (A.8) that no two calls are responded to by same ambulance if their response time frames overlap, the constraint (1.3) is also satisfied. Hence, the constructed solution $\mathbf{x}^{D-BLP}$ which is equivalent to $\mathbf{x}^{BLP}$ is feasible in our formulation. Hence we are done.

A.1.3 Proof of Proposition 1.3:

In transitioning the SBROM into SBROM-MILP, we simply define an auxiliary variable:

$$temp_k = \max_{n \in \{1,\dots,N(k)\}} \sum_{c \in C} \sum_{c^2 \in C^2} \sum_{v \in B} l^{kn}_{cc^2v} x^{kn}_{cc^2v}.$$

A.1.4 Integer Program to Generate New Subset of Scenarios

Let us denote the upper bound at each iteration with $\bar{\nu}_\tau$. We denote the number of current iteration with L. Let $\phi_{k,s}$ and $\mu_{\tau,k,s}$ be the decision variables where k and s correspond to uncertainty set and scenario indices. Then the integer optimization problem to find the new sets is given as follows:

$$\min_{\phi_{k,s} \& \mu_{\tau,k,s}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{N}(k)} \phi_{k,s} \tag{A.10}$$

$$s.t.\ \phi_{k,s} \geq \mu_{\tau,k,s} \qquad \forall k \in \mathcal{K},\ \forall s \in \mathcal{N}(k),\ \forall \tau = 1,\dots,L \tag{A.11}$$

$$\sum_{s \in \mathcal{N}(k)} \mu_{\tau,k,s} = 1, \qquad \forall k \in \mathcal{K},\ \forall \tau = 1,\dots,L \tag{A.12}$$

$$\sum_{k \in \mathcal{K}} p_k \times \left(\sum_{s \in \mathcal{N}(k)} \mu_{\tau,k,s} Z(\tau,s,k) \right) \geq \bar{\nu}_L, \qquad \forall \tau = 1,\dots,L \tag{A.13}$$

$$\phi_{k,s}, \mu_{\tau,k,s} \in \{0,1\} \qquad \forall k \in \mathcal{K},\ \forall s \in \mathcal{N}(k),\ \forall \tau = 1,\dots,L \tag{A.14}$$

Let $\phi_{k,s}$ and $\mu_{\tau,k,s}$ denote the optimal solution. Then the new subset of scenarios for iteration $\tau + 1$ are:

$$\hat{S}^k(c)_{\tau+1} = \{s^k_s(c) : \phi_{k,s} = 1, s \in \mathcal{N}(k)\} \qquad \forall k \in \mathcal{K} \tag{A.15}$$

Appendix B

Appendices for Chapter 2

B.1 Conversion to Regime $\mathbb{C}$ from Regime $\mathbb{P}$

In order to use the effective match rate expressions for Regime $\mathbb{C}$ by simply switching the roles of providers and consumers, we transform this system to an equivalent system. We carry out this transformation by simply speeding up time by a factor of K^p, so that (i) the potential arrival rate of providers (resp. consumers) is now Λ_p/K^p (resp. Λ_c/K^p), and (ii) the multiplier for the expected delay in the utility functions (Eq. (2.2)) is $1/K^p$ for the consumers and 1 for the providers (after this transformation, K^p remains the relative value of time for the providers compared to the consumers). This system is identical to the original system, i.e., providers and consumers obtain identical utilities at each state (relative to the original system), but where time is sped up by a factor of K^p. Accordingly, the match rate we obtain from this system must be multiplied by K^p to obtain the match rate in the original unit of time. Thus, we can use this equivalent construction to write expressions for the effective match rate in Regime $\mathbb{C}$ using the effective match rate expressions for Regime $\mathbb{P}$ by following these steps: Replace (i) Λ^c with Λ^p/K^p; (ii) Λ^p with Λ^c/K^p; (iii) t^c with t^p; (iv) t^p with t^c; (v) δ^c with δ^p; (vi) δ^p with δ^c; and (vii) K^p with $1/K^p$. Subsequently, compute the effective match rate in this transformed system and multiply the result by K^p.

B.2 Proofs of Propositions

B.2.1 Proof of Proposition 2.1

For states $n \in \{0, 1, \ldots, b^p_{\mathbb{O}} - 1\}$, providers join at instantaneous rate $\delta^p\Lambda^p$, and for states $n \leq -1$, they join at rate Λ^p. For states $n \in \{-b^c_{\mathbb{O}} + 1, \ldots, -1, 0\}$, consumers join at instantaneous rate $\delta^c\Lambda^c$, and for states $n \geq 1$, they join at rate Λ^c. The resulting state-dependent instantaneous arrival rates are tabulated in Proposition 2.1 in which the cases represent whether the bounding states are zero or greater than zero.

Eq. (2.3) gives the general expression for the effective match rate. In Regime $\mathbb{O}$, we have a truncated state-space, i.e., some terms in Eq. (2.3) are zero. We compute the effective match rates as the weighted

average of the providers' instantaneous arrival rates with the weights being the steady-state probabilities:

$$M_{\mathbb{O}} = \begin{cases} \sum_{n=-b_{\mathbb{O}}^c}^{-1} \pi_{\mathbb{O}}(n)\Lambda^p + \sum_{n=0}^{b_{\mathbb{O}}^p-1} \pi_{\mathbb{O}}(n)\delta^p\Lambda^p & \text{for } b_{\mathbb{O}}^c > 0,\ b_{\mathbb{O}}^p > 0 \\ \sum_{n=-b_{\mathbb{O}}^c}^{-1} \pi_{\mathbb{O}}(n)\Lambda^p & \text{for } b_{\mathbb{O}}^c < 0,\ b_{\mathbb{O}}^p = 0 \\ \sum_{n=0}^{b_{\mathbb{O}}^p-1} \pi_{\mathbb{O}}(n)\delta^p\Lambda^p & \text{for } b_{\mathbb{O}}^c = 0,\ b_{\mathbb{O}}^p > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.1}$$

The steady state probabilities are obtained by solving the set of balance equations and the normalization equation (check the CTMC in Fig. 2.1):

$$\begin{aligned} &\pi_{\mathbb{O}}(n) = \pi_{\mathbb{O}}(0)\left(\frac{\delta^p\Lambda^p}{\Lambda^c}\right)^n, \qquad \forall n \in \{0, 1, \ldots, b_{\mathbb{O}}^p\}, \\ &\pi_{\mathbb{O}}(n) = \pi_{\mathbb{O}}(0)\left(\frac{\delta^c\Lambda^c}{\Lambda^p}\right)^{|n|}, \qquad \forall n \in \{-b_{\mathbb{O}}^c, \ldots, -1, 0\}, \\ &\sum_{n=-b_{\mathbb{O}}^c}^{b_{\mathbb{O}}^p} \pi_{\mathbb{O}}(n) = 1. \end{aligned}$$

By solving the normalization equation we obtain $\pi_{\mathbb{O}}(0)$ as:

$$\pi_{\mathbb{O}}(0) = \begin{cases} \left(\frac{\Lambda^c - \Lambda^p\delta^p(\Lambda^p\delta^p/\Lambda^c)^{\left\lfloor \frac{\Lambda^c - t^p\Lambda^c}{K^p t^p} \right\rfloor}}{\Lambda^c - \Lambda^p\delta^p} - \frac{\Lambda^c\delta^c\left(1-(\Lambda^c\delta^c/\Lambda^p)^{\left\lfloor \left(\frac{1}{t^c}-1\right)\Lambda^p \right\rfloor}\right)}{\Lambda^c\delta^c - \Lambda^p} \right)^{-1} & \text{for } b_{\mathbb{O}}^c > 0,\ b_{\mathbb{O}}^p > 0 \\ \frac{\Lambda^p - \Lambda^c\delta^c}{\Lambda^p - \Lambda^c\delta^c(\Lambda^c\delta^c/\Lambda^p)^{\left\lfloor \left(\frac{1}{t^c}-1\right)\Lambda^p \right\rfloor}} & \text{for } b_{\mathbb{O}}^c > 0,\ b_{\mathbb{O}}^p = 0 \\ \frac{\Lambda^c - \Lambda^p\delta^p}{\Lambda^c - \Lambda^p\delta^p(\Lambda^p\delta^p/\Lambda^c)^{\left\lfloor \frac{\Lambda^c - t^p\Lambda^c}{K^p t^p} \right\rfloor}} & \text{for } b_{\mathbb{O}}^c = 0,\ b_{\mathbb{O}}^p > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.2}$$

Substituting the resulting steady-state probabilities into Eq. (B.1), we derive the match rates in Regime $\mathbb{O}$.

B.2.2 Proof of Proposition 2.2

We begin by describing the three possible joining strategies that patient consumers can employ:

- Strategy (i): All arriving patient consumers join with probability one, resulting in the instantaneous arrival rate $\delta^c\Lambda^c$. This occurs if the consumers' resulting expected delay is small enough, given their delay sensitivity parameter t^c:

$$U_{\mathbb{P}}^c(\emptyset) = R^c\left(1 - t^c\left(1 + E[W_{\mathbb{P}}^c(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = 1]\right)\right) > 0. \tag{B.3}$$

- Strategy (ii): All arriving patient consumers join with probability $j \in (0,1)$, resulting in the instantaneous arrival rate $j\delta^c\Lambda^c$. This occurs if the consumers' resulting expected delay is such that they are indifferent between joining and not joining, given their delay sensitivity parameter t^c:

$$U^c_{\mathbb{P}}(\emptyset) = R^c\left(1 - t^c\left(1 + E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = j]\right)\right) = 0. \tag{B.4}$$

- Strategy (iii): All arriving patient consumers do not join, resulting in an instantaneous arrival rate zero. This occurs if patient consumers joining with any positive probability results in a negative expected utility, given their delay sensitivity parameter t^c:

$$U^c_{\mathbb{P}}(\emptyset) = R^c\left(1 - t^c\left(1 + E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = \epsilon]\right)\right) < 0, \forall \epsilon > 0. \tag{B.5}$$

These strategies, taken together with the joining behavior of providers, result in five distinct cases:

Case 1 ($J_{\mathbb{P}}(t^c|\emptyset) = 1$ and $b^p_{\mathbb{P}} > 0$): In this case, the instantaneous arrival rate of consumers is $\lambda^c_{\mathbb{P}}(s^c_{\mathbb{P}}(n) = \emptyset) = \delta^c\Lambda^c,\ \forall n$ and the instantaneous arrival rate of providers is $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \delta^p\Lambda^p,$ for $0 \le n < b^p$ and $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \Lambda^p,$ for $n < 0$.

Case 2 ($J_{\mathbb{P}}(t^c|\emptyset) = j \in (0,1)$ and $b^p_{\mathbb{P}} > 0$): The instantaneous arrival rate of consumers is $\lambda^c_{\mathbb{P}}(s^c_{\mathbb{P}}(n) = \emptyset) = j\delta^c\Lambda^c,\ \forall n$ and the instantaneous arrival rate of providers is $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \delta^p\Lambda^p,\ \forall 0 \le n < b^p$ and $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \Lambda^p,\ \forall n < 0$.

Case 3 ($J_{\mathbb{P}}(t^c|\emptyset) = 1$ and $b^p_{\mathbb{P}} = 0$): The instantaneous arrival rate of consumers is $\lambda^c_{\mathbb{P}}(s^c_{\mathbb{P}}(n) = \emptyset) = \delta^c\Lambda^c,\ \forall n$ and the instantaneous arrival rate of providers is $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \Lambda^p,\ \forall n < 0$.

Case 4 ($J_{\mathbb{P}}(t^c|\emptyset) = j \in (0,1)$ and $b^p_{\mathbb{P}} = 0$): The instantaneous arrival rate of consumers is $\lambda^c_{\mathbb{P}}(s^c_{\mathbb{P}}(n) = \emptyset) = j\delta^c\Lambda^c,\ \forall n$ and the instantaneous arrival rate of providers is $\lambda^p_{\mathbb{P}}(s^p_{\mathbb{P}}(n) = n) = \Lambda^p,\ \forall n < 0$.

Case 5 ($J_{\mathbb{P}}(t^c|\emptyset) = 0$): If none of the above cases holds, patient consumers join the system with probability $J_{\mathbb{P}}(t^c|\emptyset) = 0$ (this automatically implies that the bounding state for providers is zero, i.e., $b^p_{\mathbb{P}} = 0$). In this case, no user joins the system and the system is empty.

For each case in Proposition 2.2 we show how to obtain the required conditions and the effective match rates. For cases under which the arrival rate $\lambda^c(\emptyset)$ for consumers is either $\delta^c\Lambda^c$ or 0 (Cases 1, 3 and 5), we use these arrival rates to find the providers' bounding state. Given the bounding state and the instantaneous arrival rates of both user classes, we solve for the stationary probabilities of the CTMC, from which we obtain the expected delay for an arriving consumer using Eq. (2.12). We use this expected delay to find the parameter conditions that are consistent with the specific case. For cases under which consumers employ a mixed strategy (Cases 2 and 4), we use an alternative approach.

Case 1: We re-arrange the expression (2.10), substituting $\lambda^c_{\mathbb{P}}(\emptyset) = \delta^c\Lambda^c$ to obtain the required condition on t^p.

$$1 \le \frac{\delta^c\Lambda^c\left(1 - t^p\right)}{K^p t^p} \implies t^p \le \frac{\delta^c\Lambda^c}{\delta^c\Lambda^c + K^p}$$

Next, we substitute $\lambda^c_{\mathbb{P}}(\emptyset) = \delta^c\Lambda^c$ and $b^p_{\mathbb{P}} = \left\lfloor \frac{\delta^c\Lambda^c\left(1 - t^p\right)}{t^p K^p} \right\rfloor$ to find the probabilities π_i for the CTMC in Fig. 2.2. Note that in order for the resulting steady state probabilities to be valid, the CTMC must be stable, i.e. $\Lambda^p > \Lambda^c\delta^c$. Otherwise, the arrival rate of consumers is too high for the chain to be stable, leading to an unbounded expected delay for the consumers. Accordingly Case 1 requires that $\Lambda^p > \Lambda^c\delta^c$. Under this

condition, we use the steady state probabilities π_i to calculate $E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset)=1]$ using Eq. (2.12). We then substitute this into expression (B.3) to obtain the following condition on t^c.

$$t^c \leq \frac{1}{1+\frac{\delta^c\Lambda^c-\delta^p\Lambda^p}{(\Lambda^p-\delta^c\Lambda^c)\left(\delta^c\Lambda^c(1-\delta^p)-\delta^p(\Lambda^p-\delta^c\Lambda^c)\left(\frac{\Lambda^p\delta^p}{\delta^c\Lambda^c}\right)^{b^p_{\mathbb{P}}}\right)}} \qquad \text{where} \qquad b^p_{\mathbb{P}} = \left\lfloor \frac{\delta^c\Lambda^c\,(1-t^p)}{t^pK^p} \right\rfloor. \tag{B.6}$$

By substituting $H^c = t^c/(1-t^c)$ and $H^p = t^pK^p/(1-t^p)$ we obtain the required condition. The resulting match rate is equal to $\lambda^c_{\mathbb{P}}(\emptyset) = \delta^c\Lambda^c$.

Case 2: We first derive the conditions on t^c and t^p that are consistent with Case 2. Since the joining decisions of consumers and providers are intertwined, we derive conditions for Case 2 by considering the behavior of consumers and providers together. In order for Case 2 to occur, consumers must join the system with a probability less than 1, i.e., their patience level is too small for them to join the system with probability 1. This either occurs when t^c is larger than the upper bound on t^c presented for Case 1 or when Case 1 results in an unstable system. In the former case, the upper bound on t^c presented for Case 1 forms a lower bound on t^c (which in turn depends on t^p through the bounding state $b^p_{\mathbb{P}}$ as in Eq. (B.6)). In the latter case, consumers join the system with a probability less than 1 because the underlying CTMC is not stable under Case 1, i.e., $\Lambda^p \leq \Lambda^c\delta^c$.

Now, we describe how to find an upper bound on t^c and t^p. As consumers become more impatient (as t^c increases from its lower bound), consumers join with a lower rate $\lambda^c_{\mathbb{P}}(\emptyset)$, leading to a smaller bounding state $b^p_{\mathbb{P}}$ for the providers' side. Therefore, the upper bound on t^c corresponds to the lowest possible value of $b^p_{\mathbb{P}}$ that it induces that is consistent with Case 2, i.e., $b^p_{\mathbb{P}} = 1$. When the bounding state is 1, we can find the expected delay $E[W^c_1(\emptyset)|\lambda^{t^c}_{\mathbb{P}}(\emptyset)]$, and the arrival rate $\lambda^{t^c}_{\mathbb{P}}(\emptyset)$ that are consistent with each other in closed form by solving the system of equations (B.7) -(B.8) for the arrival rate l:

$$E[W^c_1(\emptyset)|\lambda^{t^c}_{\mathbb{P}}(\emptyset)=l] = \frac{l}{(\Lambda^p-l)(l(1-\delta^p)+\Lambda^p\delta^p)} \tag{B.7}$$

$$(1-t^c(1+E[W^c_1(\emptyset)|\lambda^{t^c}_{\mathbb{P}}(\emptyset)=l])) = 0 \tag{B.8}$$

Solving the above equations, we obtain:

$$l = \frac{\Lambda^p\left(-\sqrt{K^{p2}t^{c2}-2\Lambda^cK^p(1-2\delta^p)(1-t^c)t^c+\Lambda^{c2}(1-t^c)^2}-K^pt^c+\Lambda^c(1-2\delta^p)(1-t^c)\right)}{2\Lambda^c(1-\delta^p)(1-t^c)}. \tag{B.9}$$

Then, we ensure that providers are patient enough to join at state 0 when consumers join at rate l. (If they are, the resulting l is strictly positive.) From Eq. (2.10), we obtain this condition as:

$$t^p \leq \frac{l}{K^p+l}, \qquad \text{where } l \text{ is as given in Eq. (B.9)} \tag{B.10}$$

This condition can equivalently be formulated as an upper bound on t^c, in terms of t^p; this version is provided in Eq. (B.11).

$t^c \leq$

$$\frac{(K^p t^p - \Lambda^p + \Lambda^p t^p)(K^p \delta^p t^p - K^p t^p - \Lambda^p \delta^p + \Lambda^p \delta^p t^p)}{K^{p^2} \delta^p t^{p^2} - 2K^{p^2} t^{p^2} + K^{p^2} t^p + 2\Lambda^p K^p \delta^p t^{p^2} - 2\Lambda^p K^p \delta^p t^p - \Lambda^p K^p t^{p^2} + \Lambda^p K^p t^p + \Lambda^{p^2} \delta^p + \Lambda^{p^2} \delta^p t^{p^2} - 2\Lambda^{p^2} \delta^p t^p} \tag{B.11}$$

By substituting $H^c = t^c/(1-t^c)$ and $H^p = t^p K^p/(1-t^p)$ we obtain the required condition.

Although we can obtain the conditions under which Case 2 holds in closed form, we cannot find the resulting effective match rate in closed form. This is because the consumers' expected delay $E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = j]$ depends on both the consumers' instantaneous arrival rate at every state $\lambda^c_{\mathbb{P}}(\emptyset) = j\delta^c\lambda^c$ and the providers' bounding state. However, in order to find the equilibrium joining strategy of consumers, we must find the value of $J_{\mathbb{P}}(t^c|\emptyset)$ that solves Eq. (B.4). This equation cannot, in general, be solved in closed-form, because the expression for the expected delay $E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset)]$ contains the providers' bounding state b^p in the exponent, in which the unknown $J_{\mathbb{P}}(t^c|\emptyset)$ appears inside a floor function as well as outside the exponent. Note that the Eq. (2.14) is what must be solved to obtain consumers' instantaneous arrival rate; the result corresponds to the effective match rate of the system.

Case 3: Under this case, the system reduces to the well-known $M/M/1$ queuing system with arrival rate $\delta^c\Lambda^c$ and service rate Λ^p. Similar to Case 1, the first condition for this case is the stability of the CTMC, i.e., $\Lambda^p > \delta^c\Lambda^c$. If this condition is violated, the expected delay for consumers is unbounded, leading to Case 3 not holding (consumers must join with some probability less than one to ensure a bounded delay). Given that the system is stable, we use off-the-shelf closed form expressions for expected delay in an $M/M/1$. We re-arrange Eq. (2.10) to obtain a lower bound on the providers' patience level, ensuring that the resulting bounding state is 0 in equilibrium:

$$\frac{\delta^c\Lambda^c\,(1-t^p)}{K^p t^p} < 1 \implies t^p > \frac{\delta^c\Lambda^c}{\delta^c\Lambda^c + K^p}$$

Similarly, we re-arrange Eq. (B.3) to obtain an upper bound on the consumers' patience level, ensuring that they join at rate $\delta^c\Lambda^c$ in equilibrium:

$$t^c \leq \frac{\Lambda^p - \delta^c\Lambda^c}{1 + \Lambda^p - \delta^c\Lambda^c}$$

By substituting $H^c = t^c/(1-t^c)$ and $H^p = t^p K^p/(1-t^p)$ we obtain the required condition. As with Case 1, the resulting effective match rate is $\lambda^c_{\mathbb{P}}(\emptyset) = \delta^c\Lambda^c$.

Case 4: Under this case, note that consumers' expected delay $E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = j]$ is dependent on their own instantaneous arrival rate $\lambda^c_{\mathbb{P}}(\emptyset) = j\delta^c\lambda^c$ but not dependent on the providers' bounding state (which is 0 in Case 4). Accordingly, in order to find the equilibrium joining strategy of consumers, we must find the value of $J_{\mathbb{P}}(t^c|\emptyset)$ that solves Eq. (B.4). Fortunately, unlike in Case 2, since the bounding state for providers is 0, the system simplifies to an $M/M/1$ queuing system with an arrival rate of $\lambda^c_{\mathbb{P}}(\emptyset)$ and a service rate of Λ^p. Hence, the expected delay in terms of the unknown consumers' instantaneous arrival rate can be easily found to be $E[W^c_{\mathbb{P}}(\emptyset)|J_{\mathbb{P}}(t^c|\emptyset) = j] = \dfrac{1}{\Lambda^p - j\delta^c\Lambda^c}$. Then, using Eq. (B.4) we solve for the consumers' equilibrium joining probability j and obtain the consumers' arrival rate as $\lambda^c_{\mathbb{P}}(\emptyset) = \Lambda^p - \dfrac{t^c}{1-t^c}$. In order for Case 4 to be valid, providers must be unwilling to wait in the system. Using the derived consumers' arrival rate, we derive

this condition using Eq. (2.10) to obtain:

$$\frac{(\Lambda^p - \frac{t^c}{1-t^c})(1-t^p)}{K^p t^p} < 1 \implies t^p > \frac{\Lambda^p(1-t^c) - t^c}{K^p(1-t^c) + \Lambda^p(1-t^c) - t^c}$$

Next, we must ensure that that consumers' joining probability is between 0 and 1. As explained in Case 3, this could occur if the arrival rate of patient consumers is higher than that of providers, i.e. $\Lambda^p \leq \delta^c \Lambda^c$, but the remaining conditions for Case 3 hold. If this is not the case (i.e., $\Lambda^p > \delta^c \Lambda^c$), consumers join with probability between 0 and 1 if the arrival rate expression $\Lambda^p - \frac{t^c}{1-t^c}$ yields an arrival rate for consumers that is between 0 and $\delta^c \Lambda^c$:

$$0 < \lambda^c_{\mathbb{P}}(\emptyset) = \Lambda^p - \frac{t^c}{1-t^c} < \delta^c \Lambda^c \implies \frac{\Lambda^p - \delta^c \Lambda^c}{1 + \Lambda^p - \delta^c \Lambda^c} < t^c < \frac{\Lambda^p}{1+\Lambda^p}$$

By substituting $H^c = t^c/(1-t^c)$ and $H^p = t^p K^p/(1-t^p)$ we obtain the required condition. As discussed above, the expected match rate in Case 4 is equal to $\lambda^c_{\mathbb{P}}(\emptyset) = \Lambda^p - \frac{t^c}{1-t^c}$.

Case 5: In Cases 1-4, we presented conditions under which users are willing to join with positive probability. If none of these cases applies, then we conclude that users are not willing to join the system and hence the system is empty, leading to an effective match rate of 0.

B.2.3 Proof of Proposition 2.3

We first show that the only two possible case overlaps are between Case 1 and Case 4, and between Case 2 and Case 4. To do so, we first consider every possible pair of cases below, and discuss whether they can overlap:

Case 1 and Case 2: Note that the conditions for t^c are negations of each other, hence there doesn't exist any t^c that satisfies both of the conditions.

Case 1 and Case 3: Note that the conditions for t^p are negations of each other, hence there doesn't exist any t^p that satisfies both of the conditions.

Case 2 and Case 3: Note that the RHS of the condition for t^p under Case 2 is increasing in l, i.e., $\frac{l}{K^p+l}$ is increasing in l. Also note that l is bounded above by $\delta^c \Lambda^c$, i.e., $l \leq \delta^c \Lambda^c$. Accordingly, the RHS of the t^p condition for Case 2 is smaller than the RHS of the t^p condition for Case 3. Hence, there does not exist any t^p value that satisfy both conditions.

Case 3 and Case 4: Note that the upper bound for t^c for Case 3 is the lower bound for t^c for Case 4. Hence there does not exist any t^c value that satisfies both of the conditions.

Therefore, the possible candidates for overlap are Cases 1 and 4 and Cases 2 and 4. Both these overlaps are possible, as shown in Fig. 2.3, which uses parameters $\Lambda^c = 9, \Lambda^p = 8.5, K^p = 1.5\delta^c = 0.75$ and $\delta^p = 0.9$.

Now, we compare the effective matching rate between the overlapping cases. First, note that, under Case 4 patient consumers join with probability $J_{\mathbb{P}}(t^c|\emptyset) = j \in (0,1)$ while under case 1 they join with probability $J_{\mathbb{P}}(t^c|\emptyset) = 1$. Therefore, it is immediate that the effective match rate under Case 1 is higher than that of under Case 4.

Next we show that the effective match rate (equivalently, the consumers' effective arrival rate) under Case 2 is higher than that under Case 4. In particular, consider a Case 2 equilibrium in which providers join up to state $B-1$ such that bounding state is $B \geq 1$. We will show that:

(i) For a fixed arrival rate λ^c, the consumers' utility is weakly increasing in bounding state B.

(ii) For a fixed bounding state, the consumers' utility is weakly decreasing in the arrival rate λ^c.

Part (i) implies that fixing the arrival rate at the equilibrium arrival rate corresponding to Case 4 ($B = 0$), and increasing the bounding state to $B > 0$ leaves consumers with excess utility, implying that consumers' equilibrium joining rate is different from λ^c. Part (ii) then implies that in order to find the equilibrium arrival rate under Case 2, λ^c needs to increase to a higher value than the Case 4 equilibrium arrival rate.

Proof of Part (i): Showing Part (i) is equivalent to showing that for a fixed λ^c the consumers' expected delay $E[W^c_{\mathbb{P}}(\emptyset)|\lambda^c]$ decreases in the bounding state B. In order to do so, we take the first derivative of the delay under Case 2 with respect to bounding state B and show that it is non-positive. With some abuse of notation, we have:

$$E[W^c_{\mathbb{P}}(\emptyset)|\lambda^c] = \frac{\Lambda^p\delta^p - \lambda^c}{(\lambda^c - \Lambda^p)\left(\delta^p(\lambda^c - \Lambda^p)\left(\frac{\Lambda^p\delta^p}{\lambda^c}\right)^B + \lambda^c - \lambda^c\delta^p\right)}$$

$$\frac{\partial E[W^c_{\mathbb{P}}(\emptyset)|\lambda^c]}{\partial B} = -\frac{\delta^p\left(\frac{\delta^p\Lambda^p}{\lambda^c}\right)^B \overbrace{(\Lambda^p\delta^p - \lambda^c)\log\left(\frac{\delta^p\Lambda^p}{\lambda^c}\right)}^{>0}}{\left(\delta^p(\lambda^c - \Lambda^p)\left(\frac{\Lambda^p\delta^p}{\lambda^c}\right)^B + \lambda^c - \lambda^c\delta^p\right)^2} \leq 0$$

Proof of Part (ii): We equivalently show that for a fixed bounding state B, the consumers' expected delay $E[W^c_{\mathbb{P}}(\emptyset)|\lambda^c]$ increases in their arrival rate λ^c. Observe that the CTMC in Fig. 2.2 is a birth-death process. As a direct consequence of Theorem 5 in Smith and Whitt (1981) we have that:

$$\Pr(i \leq j) \text{ is increasing in } \lambda^c,\ \forall j, \tag{B.12}$$

where i and j are state indices corresponding to the CTMC in Fig. 2.2. (In particular, the result in Smith and Whitt (1981) implies a likelihood ratio ordering between the Markov chain with a lower λ^c and that with a higher λ^c; this in turn implies the first-order stochastic dominance in Eq. (B.12).) From Eq. (B.12), we have:

$$\begin{aligned}
&\sum_{j=-\infty}^{0} \Pr(i \leq j) \text{ is increasing in } \lambda^c \\
\Leftrightarrow &\sum_{j=-\infty}^{0} (i+1)\Pr(i = j) \text{ is increasing in } \lambda^c \\
\Leftrightarrow &\sum_{j=-\infty}^{0} \frac{i+1}{\Lambda^p}\Pr(i = j) \text{ is increasing in } \lambda^c \\
\Leftrightarrow &E[W^c_{\mathbb{P}}(\emptyset)|\lambda^c] \text{ is increasing in } \lambda^c.
\end{aligned}$$

This completes the proof.

B.2.4 Proof of Proposition 2.5:

We begin by proving the conditions for Regime $\mathbb{P}$'s optimality. Given that the Case 1 conditions hold for Regime $\mathbb{P}$, the effective match rate under Regime $\mathbb{P}$ is $\delta^c\Lambda^c$ and effective match rate under Regime $\mathbb{C}$ is bounded above by $\delta^p\Lambda^p$. Accordingly, the first and trivial condition for Regime $\mathbb{P}$ to yield a higher match

rate than Regime $\mathbb{C}$ is that $\delta^c\Lambda^c > \delta^p\Lambda^p$.

Next, we derive conditions under which the match rate under Regime $\mathbb{O}$ is lower than $\delta^c\Lambda^c$. We we write the match rate for Regime $\mathbb{O}$ in general form:

$$M_{\mathbb{O}} = \sum_{n=-b^c_{\mathbb{O}}+1}^{0} \delta^c\Lambda^c\pi_{\mathbb{O}}(n) + \sum_{n=1}^{b^p_{\mathbb{O}}} \Lambda^c\pi_{\mathbb{O}}(n),$$

where the first term vanishes if $b^c_{\mathbb{O}} = 0$ and the second term vanishes if $b^p_{\mathbb{O}} = 0$. Using the fact that $\pi_{\mathbb{O}}(-b^c_{\mathbb{O}}) + \sum_{n=-b^c_{\mathbb{O}}+1}^{0} \pi_{\mathbb{O}}(n) + \sum_{n=1}^{b^p_{\mathbb{O}}} \pi_{\mathbb{O}}(n) = 1$, algebraic manipulation yields:

$$\delta^c\Lambda^c\pi_{\mathbb{O}}(-b^c_{\mathbb{O}}) + M_{\mathbb{O}} - (1-\delta^c)\Lambda^c \sum_{n=1}^{b^p_{\mathbb{O}}} \pi_{\mathbb{O}}(n) = \delta^c\Lambda^c = M_{\mathbb{P}}. \tag{B.13}$$

Eq. (B.13) allows us to derive conditions for comparing the match rates under Regimes $\mathbb{P}$ and $\mathbb{O}$. More specifically, we derive that Regime $\mathbb{P}$ yields higher match rates than Regime $\mathbb{O}$ when:

$$\delta^c\Lambda^c\pi_{\mathbb{O}}(-b^c_{\mathbb{O}}) > (1-\delta^c)\Lambda^c \sum_{n=1}^{b^p_{\mathbb{O}}} \pi_{\mathbb{O}}(n). \tag{B.14}$$

Substituting the value of the stationary probabilities and the bounding states, Ineq. (B.14) becomes:

$$\delta^c \left(\frac{\Lambda^c\delta^c}{\Lambda^p}\right)^{\lfloor(\frac{1}{t^c}-1)\Lambda^p\rfloor} > \frac{\Lambda^p(\delta^c-1)\delta^p\left(1-\left(\frac{\Lambda^p\delta^p}{\Lambda^c}\right)^{\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^pt^p}\rfloor}\right)}{\Lambda^p\delta^p-\Lambda^c}. \tag{B.15}$$

The proof for Regime $\mathbb{C}$ follows very similar steps to the proof for Regime $\mathbb{P}$. For that reason we just show the expressions that change below.

First, it is trivial that the first condition needed for Regime $\mathbb{C}$ to be optimal is $\delta^p\Lambda^p > \delta^c\Lambda^c$.

The inequality corresponding to Ineq. (B.14) becomes:

$$\Lambda^p\delta^p \left(\frac{\Lambda^p\delta^p}{\Lambda^c}\right)^{\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^pt^p}\rfloor} > \frac{\Lambda^c\Lambda^p\delta^c(\delta^p-1)\left(1-\left(\frac{\Lambda^c\delta^c}{\Lambda^p}\right)^{\lfloor(\frac{1}{t^c}-1)\Lambda^p\rfloor}\right)}{\Lambda^c\delta^c-\Lambda^p} \tag{B.16}$$

Then for Regime $\mathbb{O}$ to be optimal we simply reverse the conditions presented in ineqs. (B.15) and (B.16).

B.2.5 Proof of Proposition 2.6

We prove only part (i) of Proposition 2.6; the proof of part (ii) is symmetric to that of part (i) with consumers and providers switching roles. We will first show that when $t^p \in (\frac{\Lambda^c}{K^p+\Lambda^c}, 1)$, the effective match rates under Regimes $\mathbb{P}$ and $\mathbb{O}$ are bounded above by $\delta^c\Lambda^c$. Subsequently, we will show that when $t^c \to 0$ $\delta^c\Lambda^c < \delta^p\Lambda^p$, the effective match rate under Regime $\mathbb{C}$ is greater than $\delta^c\Lambda^c$.

We begin with Regimes $\mathbb{P}$ and $\mathbb{O}$. Note that under both regimes, the providers receive information and that the patient users' delay sensitivity is large i.e., $t^p \in (\frac{\Lambda^c}{K^p+\Lambda^c}, 1)$. Under this setting, based on (2.6), the bounding state for providers in both regimes is 0. Accordingly, in both regimes, impatient consumers do not

join the system as there will always be a delay for consumers in the system. Hence, only patient consumers can potentially join the system and accordingly, the patient consumers' maximum arrival rate $\delta^c\Lambda^c$ becomes the best possible match rate under Regimes $\mathbb{P}$ and $\mathbb{O}$

Next we show that the match rate under Regime $\mathbb{C}$ is greater than $\delta^c\Lambda^c$. We show this by examining the utility of providers when they join with rate $\delta^c\Lambda^c$ (i.e., with probability $J_{\mathbb{P}(t^p|\emptyset)} = \frac{\delta^c\Lambda^c}{\delta^p\Lambda^p} < 1$): if their utility is greater than 0 when they join with this probability, the equilibrium joining probability will be higher than $\frac{\delta^c\Lambda^c}{\delta^p\Lambda^p}$. Lastly, the providers' arrival rate is the effective match rate of the system under Regime $\mathbb{C}$.

Using the Markov Chain for Regime $\mathbb{C}$ and replacing the providers' arrival rate by $\delta^c\Lambda^c$, we obtain expressions for the steady state probabilities, the resulting expected delay and providers' expected utility as follows:

$$\pi_{\mathbb{C}}(0) = \frac{\delta^c - 1}{\delta^c b^c - b^c - 1} \tag{B.17}$$

$$\mathbb{E}[W^p_{\mathbb{C}}(\emptyset)] = \frac{1}{\Lambda^c(\delta^c - 1)\left(\delta^c b^c - b^c - 1\right)} \tag{B.18}$$

$$\mathbb{E}[U^p_{\mathbb{C}}(\emptyset)] = 1 - t^p\left(\frac{K^p}{\Lambda^c(\delta^c - 1)\left(\delta^c b^c - b^c - 1\right)} + 1\right), \tag{B.19}$$

where $b^c = \left\lfloor \frac{\delta^c\Lambda^c(1-t^c)}{t^c} \right\rfloor$.

Note that delay (B.18) is decreasing in b^c (and utility (B.19) is increasing in b^c), and b^c contains a floor function. Accordingly, replacing b^c by $\frac{\delta^c\Lambda^c(1-t^c)}{t^c} - 1$ yields an upper bound on the delay (B.18), and hence a lower bound on the utility (B.19):

$$\mathbb{E}[U^p_{\mathbb{C}}(\emptyset)] \geq 1 - t^p\left(\frac{K^p}{\Lambda^c(\delta^c - 1)\left(\delta^c\left(\frac{\Lambda^c\delta^c(1-t^c)}{t^c} - 1\right) - \left(\frac{\Lambda^c\delta^c(1-t^c)}{t^c} - 1\right) - 1\right)} + 1\right).$$

We now show that this lower bound on the utility is larger than 0 by taking its limit when t^c approaches 0 from above:

$$\lim_{t^c \to 0^+} 1 - t^p\left(\frac{K^p}{\Lambda^c(\delta^c - 1)\left(\delta^c\left(\frac{\Lambda^c\delta^c(1-t^c)}{t^c} - 1\right) - \left(\frac{\Lambda^c\delta^c(1-t^c)}{t^c} - 1\right) - 1\right)} + 1\right) = 1 - t^p > 0$$

This completes the proof.

B.2.6 Proof of Lemma 2.7:

We show for each information regime how to obtain the limiting match rate.

Under Regime $\mathbb{O}$, consumers receive occupancy information and hence their behavior is independent of other consumers' behavior. Accordingly, an unbounded increase in the arrival rate of consumers does not impact whether consumers join the system at state 0 or not. Furthermore, patient providers are incentivized to join the system at state 0 due to negligible delays. Accordingly, the bounding state for providers is guaranteed to be positive. This leaves us with only Cases 1 and 3, depending on the delay sensitivity of consumers. If the delay sensitivity of consumers is small enough, i.e., the Case 1 conditions for t^c hold in Proposition 2.1, the bounding state for both consumers and providers are positive, and if the delay sensitivity

of consumers is sufficiently large, i.e., the Case 3 conditions for t^c hold in Proposition 2.1, the bounding state for consumers is 0.

Under the first parameter setting, which occurs when $t^c \leq \frac{\Lambda^p}{1+\Lambda^p}$, the arrival rate of consumers increasing unboundedly causes the probability of system being in the bounding state for consumers to approach 1:

$$\lim_{\Lambda^c \to \infty} \pi_{\mathbb{O}}(b^c_{\mathbb{O}}) = \frac{\Lambda^{p-\left\lfloor\left(\frac{1}{t^c}-1\right)\Lambda^p\right\rfloor}\left(\Lambda^c\delta^c\right)^{\left\lfloor\left(\frac{1}{t^c}-1\right)\Lambda^p\right\rfloor}}{\frac{\Lambda^c-\Lambda^p\delta^p\Lambda^{c-\left\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^p t^p}\right\rfloor}\left(\Lambda^p\delta^p\right)^{\left\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^p t^p}\right\rfloor}}{\Lambda^c-\Lambda^p\delta^p} - \frac{\Lambda^c\delta^c\left(1-\Lambda^{p-\left\lfloor\left(\frac{1}{t^c}-1\right)\Lambda^p\right\rfloor}\left(\Lambda^c\delta^c\right)^{\left\lfloor\left(\frac{1}{t^c}-1\right)\Lambda^p\right\rfloor}\right)}{\Lambda^c\delta^c-\Lambda^p}} = 1 \quad \text{(B.20)}$$

Note also that at the bounding state for consumers, providers join with rate Λ^p as there is no delay for providers in this state. Accordingly, under this setting the match rate of the system becomes Λ^p.

Similarly, under the second parameter setting, (which corresponds to Case 3 in Proposition 2.1, corresponding to $t^c > \frac{\Lambda^p}{1+\Lambda^p}$), we examine probability of system being at state 0, since the bounding state for consumers is 0. Indeed, probability of system being at state 0 approaches 1 as $\Lambda^c \to \infty$:

$$\lim_{\Lambda^c \to \infty} \pi_{\mathbb{O}}(0) = \frac{\Lambda^c - \Lambda^p\delta^p}{\Lambda^c - \Lambda^p\delta^p\Lambda^{c-\left\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^p t^p}\right\rfloor}\left(\Lambda^p\delta^p\right)^{\left\lfloor\frac{\Lambda^c-t^p\Lambda^c}{K^p t^p}\right\rfloor}} = 1 \quad \text{(B.21)}$$

In this state, providers join at rate $\delta^p\Lambda^p$ due positive delays. Accordingly, in this setting , the match rate of the system becomes $\delta^p\Lambda^p$.

Next, we study Regime $\mathbb{C}$. As with Regime $\mathbb{O}$, consumers receive occupancy information and hence their behavior is independent of other consumers' behavior. Accordingly, an unbounded increase in the arrival rate of consumers does not impact whether consumers join the system or not. However, the unbounded increase in the arrival rate of consumers leads to negligible delays for providers (as both $\delta^c\Lambda^c$ and Λ^c are unbounded), which leads to patient providers always joining the system at their maximum possible rate, i.e., at rate $\delta^p\Lambda^p$. In more technical terms, due to negligible delays, the system is in either Case 1 or Case 3 of Proposition 2.4. Both cases yield an effective match rate of $\delta^p\Lambda^p$. Hence the effective match rate becomes $\delta^p\Lambda^p$ under Regime $\mathbb{C}$.

Under Regime $\mathbb{P}$, consumers do not receive the state information and hence their behavior is dependent on how other consumers behave. Accordingly, an unbounded increase in the arrival rate of consumers necessarily leads to mixed strategy joining behavior where only some consumers join the system, putting the system in either Case 2 or Case 4. However, since providers see negligible delays at state 0, the bounding state $b^p_{\mathbb{P}}$ for providers is strictly positive, placing the system in Case 2. Unfortunately, a closed form solution for this arrival rate does not exist as shown in Proposition 2.2. Hence we use other methods to compare this match rate with the match rate under other regimes. First, we note that the arrival rate that results from the mixed strategy is bounded above by Λ^p (from stability considerations). Second, we check whether consumers' mixing results in a match rate lower or higher than $\delta^p\Lambda^p$. We do so by calculating the expected utility of consumers if they join with arrival rate $\delta^p\Lambda^p$. If this utility is positive, then there are more consumers willing to join the system which results in a match rate higher than $\delta^p\Lambda^p$ and vice versa. Accordingly, there exists a threshold consumer delay sensitivity $\bar{t}$ above which the match rate of consumers is less than $\delta^p\Lambda^p$ and below which the match rate of consumers is more than $\delta^p\Lambda^p$. $\bar{t}$ is found by setting utility of consumers equal to 0 when their arrival rate is $\delta^p\Lambda^p$.

$$U^c_{\mathbb{P}}(-) = R^c(1 - \bar{t}(1 + E[W^c_{\mathbb{P}}(-)|\lambda^c = \delta^p\Lambda^p])) = 0 \quad \text{(B.22)}$$

$$\implies \bar{t} = \frac{1}{1 + E[W^c_{\mathbb{P}}(-)|\lambda^c = \delta^p \Lambda^p]} = \left(1 + \frac{1}{\Lambda^p(1-\delta^p)\left((1-\delta^p)\left\lfloor \frac{\delta^p(1-t^p)\Lambda^p}{K^p t^p} \right\rfloor + 1\right)}\right)^{-1}. \tag{B.23}$$

As a result, when $t^c > \bar{t}$, $M_{\mathbb{P}} < \delta^p \Lambda^p$ and when $t^c < \bar{t}$, $M_{\mathbb{P}} > \delta^p \Lambda^p$. This completes the proof

B.2.7 Proof of Proposition 2.8

We compile together the conditions given in Lemma 2.7 to produce the comparison table in Proposition 2.8.

B.3 Users' Welfare Expressions

The users' welfare expressions for Regimes $\mathbb{O}$, $\mathbb{C}$, and $\mathbb{P}$ are presented in Tables B.1-B.3, respectively. We note that the cases presented in Tables B.1-B.3 are identical to the cases presented in Propositions 2.1, 2.2 and 2.4. (Note that as with Propositions 2.2 and 2.4, the expected utilities under Case 2 for Regimes $\mathbb{P}$ and $\mathbb{C}$ cannot be found in closed-form.)

Table B.1: Users' Welfare Under Regime $\mathbb{O}$

Case	Consumers: $E[U_O^c]$
1	$\dfrac{\delta^c R^c \left(\dfrac{\Lambda^p \left((\Lambda^c \delta^c / \Lambda^p)^{b_{\mathbb{O}}^c} - 1 \right) (t^c - (\Lambda^p - \Lambda^c \delta^c)(1-t^c)) + b_{\mathbb{O}}^c t^c (\Lambda^p - \Lambda^c \delta^c) (\Lambda^c \delta^c / \Lambda^p)^{b_{\mathbb{O}}^c}}{(\Lambda^p - \Lambda^c \delta^c)^2} + \dfrac{\Lambda^p \delta^p (1-t^c) \left(1 - (\Lambda^p \delta^p / \Lambda^c)^{b_{\mathbb{O}}^p} \right)}{\Lambda^c - \Lambda^p \delta^p} \right)}{\dfrac{\Lambda^c - \Lambda^p \delta^p (\Lambda^p \delta^p / \Lambda^c)^{b_{\mathbb{O}}^p}}{\Lambda^c - \Lambda^p \delta^p} - \dfrac{\Lambda^c \delta^c \left(1 - (\Lambda^c \delta^c / \Lambda^p)^{b_{\mathbb{O}}^c} \right)}{\Lambda^c \delta^c - \Lambda^p}}$
2	$\delta^c R^c \left(\dfrac{(\Lambda^c \delta^c)^{b_{\mathbb{O}}^c} (\Lambda^p - \Lambda^c \delta^c - b_{\mathbb{O}}^c t^c + t^c (\Lambda^c \delta^c - \Lambda^p - 1))}{(\Lambda^c \delta^c)^{b_{\mathbb{O}}^c + 1} - \Lambda^{p\, b_{\mathbb{O}}^c + 1}} - t^c \left(\dfrac{1}{\Lambda^p - \Lambda^c \delta^c} + 1 \right) + 1 \right)$
3	$\delta^c R^c \dfrac{\Lambda^p \delta^p (1-t^c) \left(\Lambda^{c b_{\mathbb{O}}^p} - (\Lambda^p \delta^p)^{b_{\mathbb{O}}^p} \right)}{\Lambda^{c b_{\mathbb{O}}^p + 1} - (\Lambda^p \delta^p)^{b_{\mathbb{O}}^p + 1}}$
4	0

Case	Providers: $E[U_O^p]$
1	$\dfrac{\delta^p R^p \left(\dfrac{\Lambda^c \delta^c (1-t^p) \left(1 - (\Lambda^c \delta^c / \Lambda^p)^{b_{\mathbb{O}}^c} \right)}{\Lambda^p - \Lambda^c \delta^c} + \dfrac{b_{\mathbb{O}}^p K^p t^p (\Lambda^c - \Lambda^p \delta^p) (\Lambda^p \delta^p / \Lambda^c)^{b_{\mathbb{O}}^p}}{(\Lambda^c - \Lambda^p \delta^p)^2} - \Lambda^c \left(1 - (\Lambda^p \delta^p / \Lambda^c)^{b_{\mathbb{O}}^p} \right) (K^p t^p - (1-t^p)(\Lambda^c - \Lambda^p \delta^p)) \right)}{\dfrac{\Lambda^c - \Lambda^p \delta^p (\Lambda^p \delta^p / \Lambda^c)^{b_{\mathbb{O}}^p}}{\Lambda^c - \Lambda^p \delta^p} + \dfrac{\Lambda^c \delta^c \left(1 - (\Lambda^c \delta^c / \Lambda^p)^{b_{\mathbb{O}}^c} \right)}{\Lambda^p - \Lambda^c \delta^c}}$
2	$\delta^p R^p \dfrac{\Lambda^c \delta^c (1-t^p) \left(\Lambda^{p b_{\mathbb{O}}^c} - (\Lambda^c \delta^c)^{b_{\mathbb{O}}^c} \right)}{\Lambda^{p b_{\mathbb{O}}^c + 1} - (\Lambda^c \delta^c)^{b_{\mathbb{O}}^c + 1}}$
3	$\dfrac{\delta^p R^p \left(b_{\mathbb{O}}^p K^p t^p (\Lambda^p \delta^p)^{b_{\mathbb{O}}^p} (\Lambda^c - \Lambda^p \delta^p) - \Lambda^c \left(\Lambda^{c b_{\mathbb{O}}^p} - (\Lambda^p \delta^p)^{b_{\mathbb{O}}^p} \right) (K^p t^p - (1-t^p)(\Lambda^c - \Lambda^p \delta^p)) \right)}{(\Lambda^c - \Lambda^p \delta^p) \left(\Lambda^{c b_{\mathbb{O}}^p + 1} - (\Lambda^p \delta^p)^{b_{\mathbb{O}}^p + 1} \right)}$
4	0

where $b_{\mathbb{O}}^p$ and $b_{\mathbb{O}}^c$ are given in (2.6).

Table B.2: Users' Welfare Under Regime $\mathbb{P}$

Case	Consumers: $E[U^c_{\mathbb{P}}]$
1	$\delta^c R^c \left(t^c \left(1 - \frac{\Lambda^c\delta^c - \Lambda^p\delta^p}{(\Lambda^p - \Lambda^c\delta^c)\left(\Lambda^c\delta^c(1-\delta^p) - \delta^p(\Lambda^p - \Lambda^c\delta^c)\left(\frac{\Lambda^p\delta^p}{\Lambda^c\delta^c}\right)^{b^p_{\mathbb{P}}}\right)} \right) + 1 \right)$
2	0
3	$\delta^c R^c \left(t^c \left(1 - \frac{1}{\Lambda^p - \Lambda^c\delta^c} \right) + 1 \right)$
4	0

Case	Providers: $E[U^p_{\mathbb{P}}]$
1	$\frac{\delta^p R^p \left((\Lambda^p - \Lambda^c\delta^c)\left(\frac{\Lambda^p\delta^p}{\Lambda^c\delta^c}\right)^{b^p_{\mathbb{P}}} (b^p_{\mathbb{P}} K^p t^p(\Lambda^c\delta^c - \Lambda^p\delta^p) + \Lambda^c\delta^c(K^p t^p - (1-t^p)(\Lambda^c\delta^c - \Lambda^p\delta^p))) \right)}{\Lambda^p(\Lambda^p\delta^p - \Lambda^c\delta^c)\left(\delta^p(\Lambda^p - \Lambda^c\delta^c)\left(\frac{\Lambda^p\delta^p}{\Lambda^c\delta^c}\right)^{b^p_{\mathbb{P}}} - \Lambda^c\delta^c(1-\delta^p)\right)} + \frac{\delta^p R^p \left(\Lambda^c\delta^c(\Lambda^p(1-\delta^p)(1-t^p)(\Lambda^c\delta^c - \Lambda^p\delta^p) - K^p t^p(\Lambda^p - \Lambda^c\delta^c))\right)}{\Lambda^p(\Lambda^p\delta^p - \Lambda^c\delta^c)\left(\delta^p(\Lambda^p - \Lambda^c\delta^c)\left(\frac{\Lambda^p\delta^p}{\Lambda^c\delta^c}\right)^{b^p_{\mathbb{P}}} - \Lambda^c\delta^c(1-\delta^p)\right)}$
2	$\frac{\delta^p R^p (m(\Lambda^p(1-\delta^p)(1-t^p)(m - \Lambda^p\delta^p) - K^p t^p(\Lambda^p - m)))}{\Lambda^p(\Lambda^p\delta^p - m)\left(\delta^p(\Lambda^p - m)\left(\frac{\Lambda^p\delta^p}{m}\right)^{\left\lfloor \frac{m}{K^p}\frac{1-t^p}{t^p}\right\rfloor} - m(1-\delta^p)\right)} + \frac{(\Lambda^p - m)\left(\frac{\Lambda^p\delta^p}{m}\right)^{\left\lfloor \frac{m}{K^p}\frac{1-t^p}{t^p}\right\rfloor}\left(K^p t^p(m - \Lambda^p\delta^p)\left\lfloor \frac{m}{K^p}\frac{1-t^p}{t^p}\right\rfloor + mK^p t^p - m(1-t^p)(m - \Lambda^p\delta^p)\right)}{\Lambda^p(\Lambda^p\delta^p - m)\left(\delta^p(\Lambda^p - m)\left(\frac{\Lambda^p\delta^p}{m}\right)^{\left\lfloor \frac{m}{K^p}\frac{1-t^p}{t^p}\right\rfloor} - m(1-\delta^p)\right)}$
3	$\delta^p R^p \frac{\Lambda^c\delta^c(1-t^p)}{\Lambda^p}$
4	0

where m is found by solving (2.14) and $b^c_{\mathbb{P}}$ is given in (2.13).

Table B.3: Users' Welfare Under Regime $\mathbb{C}$

Case	Consumers: $E[U^c_{\mathbb{C}}]$
1	$\frac{\delta^c R^c \left((\Lambda^c - \Lambda^p\delta^p)\left(\frac{\Lambda^c\delta^c}{\Lambda^p\delta^p}\right)^{b^c_{\mathbb{C}}}(b^c_{\mathbb{C}}t^c(\Lambda^p\delta^p - \Lambda^c\delta^c) + \Lambda^p\delta^p(t^c - (1-t^c)(\Lambda^p\delta^p - \Lambda^c\delta^c)))\right)}{\Lambda^c(\Lambda^c\delta^c - \Lambda^p\delta^p)\left(\delta^c(\Lambda^c - \Lambda^p\delta^p)\left(\frac{\Lambda^c\delta^c}{\Lambda^p\delta^p}\right)^{b^c_{\mathbb{C}}} - \Lambda^p(1-\delta^c)\delta^p\right)} + \frac{\delta^c R^c(\Lambda^p\delta^p(\Lambda^p\delta^p(\Lambda^c(\delta^c-1)(t^c-1)+t^c)+\Lambda^c(-\Lambda^c(\delta^c-1)\delta^c(t^c-1)-t^c)))}{\Lambda^c(\Lambda^c\delta^c-\Lambda^p\delta^p)\left(\delta^c(\Lambda^c-\Lambda^p\delta^p)\left(\frac{\Lambda^c\delta^c}{\Lambda^p\delta^p}\right)^{b^c_{\mathbb{C}}}-\Lambda^p(1-\delta^c)\delta^p\right)}$
2	$\frac{m\delta^c R^c\left((\delta^c\Lambda^c)^2\Lambda^p(1-t^c) + m\Lambda^c\Lambda^p - \delta^c\Lambda^c\Lambda^p(1-t^c)(\Lambda^c+m) - mt^c(\Lambda^c(1+\Lambda^p)-m)\right)}{\Lambda^c\Lambda^p(\Lambda^c\delta^c - m)\left(\delta^c(\Lambda^c-m)\left(\frac{\Lambda^c\delta^c}{m}\right)^{\left\lfloor\left(\frac{1}{t^c}-1\right)m\right\rfloor} + m(\delta^c-1)\right)} + \frac{m\delta^c R^c(m-\Lambda^c)\left(\frac{\Lambda^c\delta^c}{m}\right)^{\left\lfloor\left(\frac{1}{t^c}-1\right)m\right\rfloor}\left(t^c\left\lfloor\left(\frac{1}{t^c}-1\right)m\right\rfloor(\Lambda^c\delta^c - m) + \Lambda^c\Lambda^p\delta^c(t^c-1) - m(\Lambda^p(t^c-1)+t^c)\right)}{\Lambda^c\Lambda^p(\Lambda^c\delta^c-m)\left(\delta^c(\Lambda^c-m)\left(\frac{\Lambda^c\delta^c}{m}\right)^{\left\lfloor\left(\frac{1}{t^c}-1\right)m\right\rfloor} + m(\delta^c-1)\right)}$
3	$\frac{\Lambda^p\delta^c\delta^p R^c(1-t^c)}{\Lambda^c}$
4	0

Case	Providers: $E[U^p_{\mathbb{C}}]$
1	$\delta^p R^p\left(t^p\left(1 - \frac{K^p(\Lambda^c\delta^c - \Lambda^p\delta^p)}{(\Lambda^c-\Lambda^p\delta^p)\left(\delta^c(\Lambda^c-\Lambda^p\delta^p)\left(\frac{\Lambda^c\delta^c}{\Lambda^p\delta^p}\right)^{b^c_{\mathbb{C}}}-\Lambda^p(1-\delta^c)\delta^p\right)}\right)+1\right)$
2	0
3	$\delta^p R^p\left(t^p\left(1 - \frac{K^p}{\Lambda^c-\Lambda^p\delta^p}\right)+1\right)$
4	0

where m is found by solving (2.16) and $b^p_{\mathbb{C}}$ is given in (2.15).

Appendix C

Appendices for Chapter 3

C.1 Experimental Setup and Parameter Settings

We report the experimental setup used throughout the chapter. As noted in the main text, we have used a value iteration algorithm with a finite state approximation by applying an upper bound on the state variable i, denoted by $N_{\max}$. We use a convergence threshold of $\varepsilon = 0.001$, i.e., the algorithm terminates when the L^0-distance between two cost-to-go functions obtained at consecutive iterations lies within $\varepsilon = 0.001$.

Throughout the experiments, we use a polynomial function for the queuing cost:

$$g(i) = K_Q \times i^x.$$

We introduce a penalty for the last queuing state $N_{\max}$, denoted by P, i.e., $g(N_{\max}) = P \times K_Q \times (N_{\max})^x$.

We use a discrete space for risk profiles, where the corresponding risk profiles are sorted from the "least risky" one to the "most risky" one. We consider the following functional form to capture risk costs:

$$R_\phi(r) = A + B \left(\frac{\mu_\phi - \min_{\phi \in \Theta} \mu_\phi}{\min_{\phi \in \Theta} \mu_\phi} \right)^C r^D.$$

C.1.1 Setting for Figure 3.2

We consider a discount rate of $\beta = 0.6$ per unit of time. The arrival rate is $\lambda = 15$ per unit of time. The queuing cost is set to $g(i) = 20 \times i^{1.5}$ per unit of time, with a penalty $P = 10$. The risk cost is set to $5 + 75 \times \left(\frac{\mu_\phi - 10}{10}\right)^3 r^2$. We consider 8 risk profiles, with a discrete probability distribution f equal to $[0.0002, 0.0068, 0.092, 0.55, 0.32, 0.026, 0.0009, 0.00012]$. We consider 5 screening options, with service rates μ_ϕ equal to $10, 12.5, 15, 17.5, 20$ per unit of time.

To obtain Figure 3.2b, we multiply the base risk costs with 5 and obtain higher risk costs. For Figure 3.2c we modify the service rates to $2, 8.5, 15, 21.5, 28$. And to obtain Figure 3.2d, we increase λ from 15 to 19.

C.1.2 Setting for Figure 3.3

Figure 3.3 relies on a more complex experimental setting to compare the results of the multi-class policy to those of the single-class policy, with various numbers of screening options and risk profiles.

We consider a discount rate of $\beta = 0.2$ per unit of time. The arrival rate is $\lambda = 18$ per unit of time. The queuing cost is set to $g(i) = 7 \times i^{1.2}$ per unit of time, with a penalty $P = 10$.

We consider varying numbers of screening options in the set Θ as follows. We first create a menu of 20 screening options, with service rates μ_ϕ equally distributed between 10 and 20 per unit of time. When we consider a fixed number of screening options, we select a subset of these 20 options—again, with service rates equally distributed between 10 and 20 per unit of time. For instance, when we consider 2 screening options, we select options 1 and 20; when we consider 3 screening options, we select options 1, 10 and 20; when we consider 5 screening options, we select options 1, 6, 10, 14 and 20; etc.

We consider varying numbers of risk profiles by defining an underlying population and aggregating it into a fixed number of categories. Specifically, we consider a population with 120 risk "bins". Let $f^p(j)$ denote the probability of risk bin j, drawn from a triangular distribution with parameters 0, 55 and 121. We define a baseline risk cost for each of the 120 bins as follows: $R_\phi^{\text{base}}(j) = 5 + 75 \times \left(\frac{\mu_\phi - 10}{10}\right)^{1.2} j^{1.1}, \forall j = 1, \cdots, 120$.

We then use this granular representation to define the risk profiles and associated risk costs, denoted by R_ϕ^{single}. For any screening option ϕ, we define the single-class risk cost, denoted by R_ϕ^{single}. It is equal to the expected risk cost resulting from the screening of any individual randomly chosen among the 120 risk bins according to distribution f^p, i.e.:

$$R_\phi^{\text{single}} = \sum_{r=1}^{120} R_\phi^{\text{base}}(j) \times f^p(j).$$

In the multi-class setting, we first aggregate the 120 risk bins into the various classes to define the probability distribution f. Let R be the number of risk profiles to select. We partition the population of 120 bins into ρ subsets, such that each of the ρ subset includes the same number of risk bins (this is possible because, by design, 120 is a common multiplier of all the numbers of risk profiles considered). Let us denote each subset by $\mathcal{S}_r \subset \{1, \cdots, 120\}$ where $r \in \{1, .., \rho\}$. Then, the probability density function f is given by:

$$f(r) = \sum_{s \in \mathcal{S}_r} f^p(s), \quad \forall r = 1, \cdots, \rho.$$

For instance, if we select 2 risk profiles, the first 60 bins and the last 60 bins are lumped into the safest risk profile and the riskier profile, respectively, with corresponding probabilities $\sum_{r=1}^{60} f^p(r)$ and $\sum_{r=61}^{120} f^p(r)$.

We now define the risk cost of each profile as follows:

$$R_\phi(r) = \frac{\sum_{s \in \mathcal{S}_r} R_\phi^{base}(s) \times f^p(s)}{f(r)}, \qquad \forall r \in \{1, ..., \rho\}.$$

Note that, by construction, the average cost in the multi-class setting matches the cost estimate in the single-class setting, for each screening option ϕ, i.e.:

$$R_\phi^{\text{single}} = \sum_{r=1,\cdots,\rho} R_\phi(r) \times f(r).$$

C.1.3 Setting for Figures 3.6, 3.7 and 3.8

In the last part of the chapter, we consider 2 risk profiles (r_1 and r_2), 2 agent types (s_1 and s_2) and 2 screening options. The two screening options are defined by service rates of 10 and 20 per unit of time and probabilities of success of 0.3 and 0.95, respectively. Recall that, for each agent type s, τ^s denotes the

minimal probability of success for the agent to remain in the system instead of reneging; we assume here that $\tau^{s_1} = 0.2$ and $\tau^{s_2} = 0.35$. Therefore, agents of type s_1 remain in the system regardless of whether the slow or the fast screening is applied while agents of type s_2 remain in the system only if the fast screening is applied (Assumption 8). We consider a discount rate of $\beta = 0.9$ per unit of time.

In Figures 3.6 and 3.7, the arrival rate is $\lambda = 15$ per unit of time. The queuing cost is set to $g(i) = 3 \times i^2$, with a penalty $P = 20$. Each agent is of type r_1 with probability 0.7 and of type r_2 with probability 0.3. The true risk cost resulting from agents going through the system successfully are $K(s_1) = 10$ and $K(s_2) = 70$.

In Figure 3.6a, the high risk cost is obtained by multiplying $K(s)$ by 2; the high queuing cost is obtained by considering $g(i) = 6 \times i^{2.1}$ and $P = 25$. In Figure 3.6b, the high and low spreads of screening rates are obtained by changing the service rate of slow screening from 10 to 8 and 12 per unit of time, respectively.

In Figures 3.8, the arrival rate is $\lambda = 18$ per unit of time. The queuing cost is set to $g(i) = 5 \times i^2$, with a penalty $P = 20$. The true risk cost resulting from agents going through the system successfully are $K(s_1) = 20$ and $K(s_2) = 200$. As detailed in the text, the probability of agents' risk profiles is calibrated to keep the underlying population unchanged. In other words, the probability that an agent is of risk profile r_1 is given by $f(r_1) = \frac{\alpha - \sigma}{2\alpha - 1}$, where σ is the proportion of the malevolent agents.

C.2 Proofs of Statements from Section 3.3

We first prove Proposition 3.3, and then Proposition 3.2—as the proof of Proposition 3.2 uses Proposition 3.3.

C.2.1 Proof of Proposition 3.3

We assume that Assumptions 3, 4 and 5 are satisfied. Recall that the optimal policy is such that $\pi^*(i, r) =$ "fast" $\iff J^*(i, r) - \bar{\mathbf{J}}(i-1) \geq \frac{\mu_F R_F(r) - \mu_S R_S(r)}{\mu_F - \mu_S}$. It suffices to show that $(\mu_F - \mu_S)\left[J^*(i, r) - \bar{\mathbf{J}}(i-1)\right] - (\mu_F R_F(r) - \mu_S R_S(r))$ is decreasing in r_0 for all $i \geq 1$. Let $r_1, r_2 \in \Re+$ be such that $r_1 < r_2$, and show that $(\mu_F - \mu_S)\left(J^*(i, r_1) - J^*(i, r_2)\right) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2)$.

We proceed by value iteration. We consider an initial cost-to-go function J_0 such that $J_0(0) = 0$ and $J_0(i, r) = 0$ for all $i \geq 1$ and $r \in \Re+$. We iteratively update the cost-to-go functions as follows, where $\bar{\mathbf{J}}_k$ denotes the expected value of $J_k(i, r)$ across all values of r, i.e., $\bar{\mathbf{J}}_k(i) = \mathbb{E}_r[J_k(i, r)] = \int_{r \in \Re+} J_k(i, r) f(r) dr, \forall i \geq 1$.

$$J_{k+1}(i, r_0) = \frac{1}{\beta + \mu_F + \lambda}\Bigg(g(i) + \lambda J_k(i+1, r_0) + \min_{\phi \in \Theta}\left\{\mu_\phi R_\phi(r_0) + \mu_\phi \bar{\mathbf{J}}_k(i-1) + (\mu_F - \mu_\phi)\, J_k(i, r_0)\right\}\Bigg) \qquad \forall i \geq 1 \tag{C.1}$$

$$J_{k+1}(0) = \frac{1}{\beta + \mu_F + \lambda}\left(g(0) + \lambda \bar{\mathbf{J}}_k(1) + \mu_F J_k(0)\right) \tag{C.2}$$

We denote by $\Delta_k(i) = (\mu_F - \mu_S)(J_k(i, r_1) - J_k(i, r_2))$ for all $i \geq 1$. We show by induction over k that $\Delta_k(i) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2), \forall i \geq 1$. This is satisfied for $k = 0$ because $\mu_F(R_F(r_2) - R_F(r_1)) \geq \mu_S(R_S(r_2) - R_S(r_1))$, which is implied by Condition (5). We now assume that it holds for a given $k \geq 0$ and show that it holds for $k + 1$.

Let us first consider a given value of $i \geq 1$ and show that $\Delta_{k+1}(i) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2)$. We denote by π_k the policy induced by the cost-to-go function J_k. From the induction hypothesis, we know that for any i, if $\pi_k(i, r_2) =$ "fast", then $\pi_k(i, r_1) =$ "fast". Therefore, we need to consider three cases

to compute $J_{k+1}(i,r_1)$ and $J_{k+1}(i,r_2)$: (i) $\pi_k(i,r_1) =$ "fast" and $\pi_k(i,r_2) =$ "fast" (ii) $\pi_k(i,r_1) =$ "fast" and $\pi_k(i,r_2) =$ "slow" (iii) $\pi_k(i,r_1) =$ "slow" and $\pi_k(i,r_2) =$ "slow"

Before proceeding further, we note that Condition (5) implies that:

$$\begin{aligned}&\frac{\lambda+\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_F\left(R_F(r_1)-R_F(r_2)\right)-\frac{\lambda}{\beta+\mu_F+\lambda}\mu_S\left(R_S(r_1)-R_S(r_2)\right)\\&\qquad\geq \mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2)\end{aligned}\tag{C.3}$$

Case 1 $\pi_k(i,r_1) =$ "fast" and $\pi_k(i,r_2) =$ "fast"

$$\begin{aligned}\Delta_{k+1}(i)&=\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\left[g(i)+\lambda J_k(i+1,r_1)+\mu_F R_F(r_1)+\mu_F\bar{\mathbf{J}}(i-1)-g(i)-\lambda J_k(i+1,r_2)-\mu_F R_F(r_2)-\mu_F\bar{\mathbf{J}}(i)\right]\\&=\frac{\lambda}{\beta+\mu_F+\lambda}\Delta_k(i+1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\left[\mu_F\left(R_F(r_1)-R_F(r_2)\right)\right]\\&\geq\frac{\lambda}{\beta+\mu_F+\lambda}(\mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2))+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_F\left(R_F(r_1)-R_F(r_2)\right)\\&\qquad\text{per the induction hypothesis}\\&=\frac{\lambda+\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_F\left(R_F(r_1)-R_F(r_2)\right)-\frac{\lambda}{\beta+\mu_F+\lambda}\mu_S\left(R_S(r_1)-R_S(r_2)\right)\\&\geq\mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2)\qquad\text{from Equation (C.3)}\end{aligned}$$

Case 2 $\pi_k(i,r_1) =$ "fast" and $\pi_k(i,r_2) =$ "slow"

$$\begin{aligned}\Delta_{k+1}(i)&=\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\big[g(i)+\lambda J_k(i+1,r_1)+\mu_F R_F(r_1)+\mu_F\bar{\mathbf{J}}(i-1)\\&\qquad-g(i)-\lambda J_k(i+1,r_2)-\mu_S R_S(r_2)-\mu_S\bar{\mathbf{J}}(i-1)-(\mu_F-\mu_S)J_k(i,r_2)\big]\\&=\frac{\lambda}{\beta+\mu_F+\lambda}\Delta_k(i+1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\left[\mu_F R_F(r_1)-\mu_S R_S(r_2)-\underbrace{(\mu_F-\mu_S)(J_k(i,r_2)-\bar{\mathbf{J}}(i-1))}_{\leq\mu_F R_F(r_2)-\mu_S R_S(r_2)}\right]\\&\geq\frac{\lambda}{\beta+\mu_F+\lambda}(\mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2))+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_F\left(R_F(r_1)-R_F(r_2)\right)\\&\qquad\text{per the induction hypothesis}\\&=\frac{\lambda+\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_F\left(R_F(r_1)-R_F(r_2)\right)-\frac{\lambda}{\beta+\mu_F+\lambda}\mu_S\left(R_S(r_1)-R_S(r_2)\right)\\&\geq\mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2)\qquad\text{from Equation (C.3)}\end{aligned}$$

Case 3 $\pi_k(i,r_1) =$ "slow" and $\pi_k(i,r_2) =$ "slow"

$$\begin{aligned}\Delta_{k+1}(i)&=\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\big[g(i)+\lambda J_k(i+1,r_1)+\mu_S R_S(r_1)+\mu_S\bar{\mathbf{J}}(i-1)+(\mu_F-\mu_S)J_k(i,r_1)\\&\qquad-g(i)-\lambda J_k(i+1,r_2)-\mu_S R_S(r_2)-\mu_S\bar{\mathbf{J}}(i-1)-(\mu_F-\mu_S)J_k(i,r_2)\big]\\&=\frac{\lambda}{\beta+\mu_F+\lambda}\Delta_k(i+1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\Delta_k(i)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_S\left[R_S(r_1)-R_S(r_2)\right]\\&\geq\frac{\lambda+\mu_F-\mu_S}{\beta+\mu_F+\lambda}(\mu_F R_F(r_1)-\mu_F R_F(r_2)-\mu_S R_S(r_1)+\mu_S R_S(r_2))+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}\mu_S\left(R_F(r_1)-R_F(r_2)\right)\end{aligned}$$

per the induction hypothesis

$$= \frac{\lambda + \mu_F - \mu_S}{\beta + \mu_F + \lambda} \mu_F \left(R_F(r_1) - R_F(r_2)\right) - \frac{\lambda}{\beta + \mu_F + \lambda} \mu_S \left(R_S(r_1) - R_S(r_2)\right)$$

$$\geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2) \qquad \text{from Equation (C.3)}$$

Therefore, $\Delta_{k+1}(i) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2)$ for any $i \geq 1$. This shows that, for any value of k, $(\mu_F - \mu_S)(J_{k+1}(i, r_1) - J_{k+1}(i, r_2)) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2)$, for each $i \geq 1$. By taking the limit when $k \to \infty$, we obtain that $(\mu_F - \mu_S)(J^*(i, r_1) - J^*(i, r_2)) \geq \mu_F R_F(r_1) - \mu_F R_F(r_2) - \mu_S R_S(r_1) + \mu_S R_S(r_2)$, for each $i \geq 1$. □

C.2.2 Proof of Proposition 3.2

We assume that Assumptions 3 and conditions, (4) and (5) are satisfied. We denote by $\bar{\mathbf{J}}^*$ the expected value of $J^*(i, r)$ across all values of r, i.e., $\bar{\mathbf{J}}^*(i) = \mathbb{E}_r[J_k^*(i, r)] = \int_{r \in \Re+} J^*(i, r) f(r) dr, \forall i \geq 1$. It suffices to show that, for any given $r_0 \in \Re+$, $J^*(i, r_0) - \bar{\mathbf{J}}^*(i-1)$ is a non-decreasing function of i. Specifically, we show that the optimal cost-to-go function satisfies the following two properties:

$$J^*(i+1, r_0) - \bar{\mathbf{J}}^*(i) - J^*(i, r_0) + \bar{\mathbf{J}}^*(i-1) \geq 0 \qquad \forall i \geq 1 \quad \forall r_0 \in \Re+ \tag{C.4}$$

$$J^*(1, r_0) - J^*(0) \geq R_S(r_0) \quad \forall r_0 \in \Re+ \tag{C.5}$$

We proceed by value iteration. We initialize the cost-to-go function with $J_0(i, r_0) = R_S(r_0), \forall i \geq 1, r_0 \in \Re+$, and $J_0(0) = 0$. We then update it as follows, with $\bar{\mathbf{J}}_k(i) = \mathbb{E}_r[J_k(i, r)]$ for all $i \geq 1$:

$$J_{k+1}(i, r_0) = \frac{1}{\beta + \mu_F + \lambda} \big(g(i) + \lambda J_k(i+1, r_0) + \min\big\{\mu_F R_F(r_0) + \mu_F \bar{\mathbf{J}}_k(i-1); \\ \mu_S R_S(r_0) + \mu_S \bar{\mathbf{J}}_k(i-1) + (\mu_F - \mu_S) J_k(i, r_0)\big\}\big) \qquad \forall i \geq 1 \tag{C.6}$$

$$J_{k+1}(0) = \frac{1}{\beta + \mu_F + \lambda} \left(g(0) + \lambda \bar{\mathbf{J}}_k(1) + \mu_F J_k(0)\right) \tag{C.7}$$

Let us consider a given $r_0 \in \Re+$ and denote by $\Delta_k(i) = J_k(i+1, r_0) - \bar{\mathbf{J}}_k(i) - J_k(i, r_0) + \bar{\mathbf{J}}_k(i-1)$, for each $i \geq 1$. We show by induction over k that $\Delta_k(i) \geq 0$ for all $i \geq 1$ and that $J_k(1, r_0) - J_k(0) \geq R_S(r_0)$. By construction, we considered $J_0(i, r_0) = R_S(r_0)$ and $J_0(0) = 0$, so these two properties are satisfied for $k = 0$. We now assume that $\Delta_k(i) \geq 0$ for all $i \geq 1$ and $J_k(1, r_0) - J_k(0) \geq R_S(r_0)$. We will show that $\Delta_{k+1}(i) \geq 0$ for all $i \geq 1$ and that $J_{k+1}(1) - J_{k+1}(0) \geq R_S(r_0)$. We denote by π_k the optimal policy given the cost-to-go function J_k, for any value k. Specifically, $\pi_k(i, r_0) =$ "fast" if $J_k(i, r_0) - \bar{\mathbf{J}}_k(i-1) \geq \frac{\mu_F R_F - \mu_S R_S}{\mu_F - \mu_S}$, and $\pi_k(i) =$ "slow" otherwise.

We first consider $i \geq 2$ and show $\Delta_{k+1}(i, r_0) \geq 0$. From Proposition 3.3 and the induction hypothesis, we know that, for any $i \geq 1$, if $r_1 < r_2$ and $\pi_k(i, r_2) =$ "fast", then $\pi_k(i, r_1) =$ "fast" and that, for any $r \in \Re+$, if $\pi_k(i, r_0) =$ "fast", then $\pi_k(i+1, r_0) =$ "fast". In other words, there exists $\hat{r}_i$ such that $\pi_k(i, r) =$ "fast" if $r \leq \hat{r}_i$ and $\pi_k(i, r) =$ "slow" if $r > \hat{r}_i$, and that $\hat{r}_i \leq \hat{r}_{i+1}$. Mathematically, we have:

$$r \leq \hat{r}_i \iff J_k(i, r_0) - \bar{\mathbf{J}}_k(i-1) \geq \frac{\mu_F R_F(r_0) - \mu_S R_S(r_0)}{\mu_F - \mu_S} \tag{C.8}$$

Consistently, we define three cases to compute $\Delta_{k+1}(i, r_0)$: (i) $\pi_k(i, r_0) =$ "fast" and $\pi_k(i+1, r_0) =$ "fast"

(ii) $\pi_k(i,r_0)=$"slow" and $\pi_k(i+1,r_0)=$"fast" (iii) $\pi_k(i,r_0)=$"slow" and $\pi_k(i+1,r_0)=$"slow"

Let us first write out $\bar{\mathbf{J}}_{k+1}(i)$:

$$\begin{aligned}
\bar{\mathbf{J}}_{k+1}(i) &= \int_0^\infty J_{k+1}(i,r)f(r)dr = \int_0^{\hat{r}_i} J_{k+1}(i,r)f(r)dr + \int_{\hat{r}_i}^\infty J_{k+1}(i,r)f(r)dr \\
&= \frac{1}{\beta+\mu_F+\lambda}\left[\int_0^{\hat{r}_i}\left(g(i)+\lambda J_k(i,r)+\mu_F R_F+\mu_F\bar{\mathbf{J}}_k(i-1)\right)f(r)dr\right] \\
&+ \frac{1}{\beta+\mu_F+\lambda}\left[\int_{\hat{r}_i}^\infty\left(g(i)+\lambda J_k(i,r)+\mu_S R_S+\mu_S\bar{\mathbf{J}}_k(i-1)+(\mu_F-\mu_S)J(i,r)\right)f(r)dr\right] \\
&= \frac{g(i)}{\beta+\mu_F+\lambda}+\frac{\lambda}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i+1)+\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}R_F(r)f(r)dr+\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr \\
&+ \int_{\hat{r}_i}^\infty\frac{\mu_S}{\beta+\mu_F+\lambda}R_S(r)f(r)dr+\int_{\hat{r}_i}^\infty\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)f(r)dr
\end{aligned}$$

Case 1 $\pi_k(i,r_0)=$"fast" and $\pi_k(i+1,r_0)=$"fast"

$$\begin{aligned}
\Delta_{k+1}(i) &= \frac{1}{\beta+\mu_F+\lambda}\left(\underbrace{g(i+1)-2g(i)+g(i-1)}_{\geq 0 \text{ by Assumption 3}}\right)+\frac{\lambda}{\beta+\mu_F+\lambda}\left(\underbrace{J_k(i+2,r_0)-\bar{\mathbf{J}}_k(i+1)-J_k(i+1,r_0)+\bar{\mathbf{J}}_k(i)}_{\geq 0 \text{ per the induction hypothesis}}\right) \\
&+ \underbrace{\frac{\mu_F}{\beta+\mu_F+\lambda}R_F(r_0)-\frac{\mu_F}{\beta+\mu_F+\lambda}R_F(r_0)}_{=0}+\frac{\mu_F}{\beta+\mu_F+\lambda}(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1)) \\
&- \int_{\hat{r}_{i-1}}^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}R_F(r)f(r)dr+\int_{\hat{r}_{i-1}}^{\hat{r}_i}\frac{\mu_S}{\beta+\mu_F+\lambda}R_S(r)f(r)dr \\
&- \int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr-\int_{\hat{r}_i}^\infty\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)\right)f(r)dr \\
&+ \int_0^{\hat{r}_{i-1}}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)f(r)dr+\int_{\hat{r}_{i-1}}^\infty\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i-1,r)\right)f(r)dr \\
&\geq \int_0^{\hat{r}_{i-1}}\frac{\mu_F}{\beta+\mu_F+\lambda}\left[\underbrace{J_k(i,r)-\bar{\mathbf{J}}_k(i-1)-J_k(i-1,r)+\bar{\mathbf{J}}_k(i-2)}_{\geq 0 \text{ per the induction hypothesis}}\right]f(r)dr \\
&+ \int_{\hat{r}_i}^\infty\frac{\mu_S}{\beta+\mu_F+\lambda}\left[\underbrace{J_k(i,r)-\bar{\mathbf{J}}_k(i-1)-J_k(i-1,r)+\bar{\mathbf{J}}_k(i-2)}_{\geq 0 \text{ per the induction hypothesis}}\right]f(r)dr \\
&+ \frac{1}{\beta+\mu_F+\lambda}\int_{\hat{r}_{i-1}}^{\hat{r}_i}\left[-\mu_S\underbrace{\left(J_k(i-1,r)-\bar{\mathbf{J}}_k(i-2)\right)}_{\leq\frac{\mu_F R_F(r)-\mu_S R_S(r)}{\mu_F-\mu_S}}+\mu_F\underbrace{\left(J_k(i,r)-\bar{\mathbf{J}}_k(i-1)\right)}_{\geq\frac{\mu_F R_F(r)-\mu_S R_S(r)}{\mu_F-\mu_S}}-\mu_F R_F(r)+\mu_S R_S(r)\right]f(r)dr \\
&\geq 0
\end{aligned}$$

Case 2 $\pi_k(i, r_0) =$ "slow" and $\pi_k(i+1, r_0) =$ "fast"

$$\begin{aligned}
\Delta_{k+1}(i) = &\frac{1}{\beta+\mu_F+\lambda}\left(\underbrace{g(i+1)-2g(i)+g(i-1)}_{\geq 0 \text{ by Assumption 3}}\right) + \frac{\lambda}{\beta+\mu_F+\lambda}\left(\underbrace{J_k(i+2,r_0)-\bar{\mathbf{J}}_k(i+1)-J_k(i+1,r_0)+\bar{\mathbf{J}}_k(i)}_{\geq 0 \text{ per the induction hypothesis}}\right)\\
&+\frac{\mu_F}{\beta+\mu_F+\lambda}R_F(r_0)-\frac{\mu_S}{\beta+\mu_F+\lambda}R_S(r_0)+\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}(i)-\frac{\mu_F-\mu_s}{\beta+\mu_F+\lambda}J_k(i,r_0)-\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}(i-1)\\
&-\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_F R_F(r)f(r)dr+\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_S R_S(r)f(r)dr\\
&-\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr-\int_{\hat{r}_i}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)\right)f(r)dr\\
&+\int_0^{\hat{r}_{i-1}}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)f(r)dr+\int_{\hat{r}_{i-1}}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i-1,r)\right)f(r)dr
\end{aligned}$$

Note that:

$$\begin{aligned}
&\mu_F R_F(r_0)-\mu_S R_S(r_0)+\mu_F\bar{\mathbf{J}}(i)-(\mu_F-\mu_S)J_k(i,r_0)-\mu_S\bar{\mathbf{J}}(i-1)\\
&=\mu_F\left(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1)\right)+\mu_F R_F(r_0)-\mu_S R_S(r_0)-(\mu_F-\mu_S)\left(J_k(i,r_0)-\bar{\mathbf{J}}(i-1)\right)\\
&\geq \mu_F\left(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1)\right) \qquad \text{because } \pi_k(i,r_0) = \text{"slow"}
\end{aligned}$$

Therefore:

$$\begin{aligned}
\Delta_{k+1}(i) \geq &\frac{\mu_F}{\beta+\mu_F+\lambda}(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1))-\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_F R_F(r)f(r)dr+\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_S R_S(r)f(r)dr\\
&-\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr-\int_{\hat{r}_i}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)\right)f(r)dr\\
&+\int_0^{\hat{r}_{i-1}}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)f(r)dr+\int_{\hat{r}_{i-1}}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i-1,r)\right)f(r)dr\\
&\geq 0 \qquad \text{by proceeding as in Case 1}
\end{aligned}$$

Case 3 $\pi_k(i, r_0) =$ "slow" and $\pi_k(i+1, r_0) =$ "slow"

$$\begin{aligned}
\Delta_{k+1}(i) = &\frac{1}{\beta+\mu_F+\lambda}\left(\underbrace{g(i+1)-2g(i)+g(i-1)}_{\geq 0 \text{ by Assumption 3}}\right) + \frac{\lambda}{\beta+\mu_F+\lambda}\left(\underbrace{J_k(i+2,r_0)-\bar{\mathbf{J}}_k(i+1)-J_k(i+1,r_0)+\bar{\mathbf{J}}_k(i)}_{\geq 0 \text{ per the induction hypothesis}}\right)\\
&+\underbrace{\frac{\mu_S}{\beta+\mu_F+\lambda}R_S(r_0)-\frac{\mu_S}{\beta+\mu_F+\lambda}R_S(r_0)}_{=0}\\
&+\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i)-\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i+1,r_0)-\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r_0)\\
&-\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_F R_F(r)f(r)dr+\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_S R_S(r)f(r)dr\\
&-\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr-\int_{\hat{r}_i}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)\right)f(r)dr
\end{aligned}$$

$$+\int_0^{\hat{r}_{i-1}} \frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)f(r)dr+\int_{\hat{r}_{i-1}}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i-1,r)\right)f(r)dr$$

Note that:

$$\begin{aligned}
&\mu_S\bar{\mathbf{J}}_k(i)-\mu_S\bar{\mathbf{J}}_k(i-1)+(\mu_F-\mu_S)J_k(i+1,r_0)-(\mu_F-\mu_S)J_k(i,r_0)\\
&=(\mu_F-\mu_S)\left(J_k(i+1,r_0)-\bar{\mathbf{J}}_k(i)-J_k(i,r_0)+\bar{\mathbf{J}}_k(i-1)\right)+\mu_F\left(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1)\right)\\
&\geq \mu_F\left(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1)\right) \qquad \text{per the induction hypothesis}
\end{aligned}$$

Therefore:

$$\begin{aligned}
\Delta_{k+1}(i)\geq\ &\frac{\mu_F}{\beta+\mu_F+\lambda}(\bar{\mathbf{J}}_k(i)-\bar{\mathbf{J}}_k(i-1))-\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_F R_F(r)f(r)dr+\int_{\hat{r}_{i-1}}^{\hat{r}_i}\mu_S R_S(r)f(r)dr\\
&-\int_0^{\hat{r}_i}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)f(r)dr-\int_{\hat{r}_i}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-1)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i,r)\right)f(r)dr\\
&+\int_0^{\hat{r}_{i-1}}\frac{\mu_F}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)f(r)dr+\int_{\hat{r}_{i-1}}^{\infty}\left(\frac{\mu_S}{\beta+\mu_F+\lambda}\bar{\mathbf{J}}_k(i-2)+\frac{\mu_F-\mu_S}{\beta+\mu_F+\lambda}J_k(i-1,r)\right)f(r)dr\\
&\geq 0 \qquad \text{by proceeding as in Case 1}
\end{aligned}$$

This completes the proof for any $i\geq 2$. We now proceed to show that $J_{k+1}(1,r_0)-J_{k+1}(0)\geq R_S(r_0)$ for all $r_0\in\Re+$. Per induction, we consider two cases: Case 1 $\pi_k(1,r_0)=$"fast" Case 2 $\pi_k(1,r_0)=$"slow"

Case 1 $\pi_k(1,r_0)=$"fast"

$$\begin{aligned}
&J_{k+1}(1,r_0)-J_{k+1}(0)\\
&=\frac{1}{\beta+\mu_F+\lambda}\left[(g(1)-g(0))+\lambda\underbrace{\left(J_k(2,r_0)-\bar{\mathbf{J}}_k(1)\right)}_{\geq J_k(1,r_0)-\bar{\mathbf{J}}_k(0)}+\mu_F R_F(r_0)\right]\\
&\geq\frac{1}{\beta+\mu_F+\lambda}\left[\underbrace{(g(1)-g(0))}_{\geq\beta R_S(r_0)\ \text{(Assumption 3)}}+\lambda R_S(r_0)+\mu_F R_F(r_0)\right] \qquad \text{per the induction hypothesis}\\
&\geq R_S(r_0)
\end{aligned}$$

Case 2 $\pi_k(1,r_0)=$"slow"

$$\begin{aligned}
&J_{k+1}(1,r_0)-J_{k+1}(0)\\
&=\frac{1}{\beta+\mu_F+\lambda}\left[(g(1)-g(0))+\lambda\underbrace{\left(J_k(2,r_0)-\bar{\mathbf{J}}_k(1)\right)}_{\geq J_k(1,r_0)-J\bar{\mathbf{J}}_k(0)}+\mu_S R_S(r_0)+(\mu_F-\mu_S)\left(J_k(1)-J_k(0)\right)\right]\\
&=\frac{1}{\beta+\mu_F+\lambda}\left[\underbrace{(g(1)-g(0))}_{\geq\beta R_S(r_0)\ \text{(Assumption 3)}}+\mu_S R_S(r_0)+(\lambda+\mu_F-\mu_S)R_S(r_0)\right] \text{ per the induction hypothesis}\\
&\geq R_S(r_0)
\end{aligned}$$

Last, we show that $\Delta_{k+1}(1, r_0) \geq 0$. Per the induction hypothesis, we consider three different cases: Case 1 $\pi_k(2, r_0) =$ "fast", $\pi_k(1, r_0) =$ "fast", Case 2 $\pi_k(2, r_0) =$ "fast", $\pi_k(1, r_0) =$ "slow", Case 3 $\pi_k(2, r_0) =$ "slow", $\pi_k(1, r_0) =$ "slow".

Before proceeding, note that for each $r \leq \hat{r}_1$, we have $J_k(1, r) - J_k(0) \geq \frac{\mu_F R_F(r) - \mu_S R_S(r)}{\mu_F - \mu_S} \geq R_F(r)$.

Case 1 $\pi_k(2, r_0) =$ "fast", $\pi_k(1, r_0) =$ "fast"

$$\begin{aligned}
\Delta_{k+1}(1, r_0) = &\frac{1}{\beta + \mu_F + \lambda} \left(\underbrace{(g(2) - 2g(1) + g(0))}_{\geq 0 \text{ by Assumption 3}} + \lambda \underbrace{(J_k(3, r_0) - \bar{\mathbf{J}}_k(2) - J_k(2, r_0) + \bar{\mathbf{J}}_k(1))}_{\geq 0 \text{ per the induction hypothesis}} \right) \\
&+ \frac{1}{\beta + \mu_F + \lambda} \left(\underbrace{-\mu_F J_k(0) + \mu_F J_k(0) + \mu_F R_F(r_0) - \mu_F R_F(r_0)}_{=0} + \mu_F \bar{\mathbf{J}}_k(1) \right) \\
&+ \frac{1}{\beta + \mu_F + \lambda} \left(- \int_0^{\hat{r}_1} (\mu_F R_F(r) + \mu_F J_k(0)) f(r) dr - \int_{\hat{r}_1}^{\infty} (\mu_S R_S(r) + \mu_S J_k(0) + (\mu_F - \mu_S) J_k(1, r)) f(r) dr \right) \\
&\geq \frac{1}{\beta + \mu_F + \lambda} \left(\mu_F \int_0^{\hat{r}_1} \underbrace{(J_k(1, r) - J_k(0) - R_F(r))}_{\geq 0} f(r) dr + \mu_S \int_{\hat{r}_1}^{\infty} \underbrace{(J_k(1, r) - J_k(0) - R_S(r))}_{\geq 0 \text{ per the induction hypothesis}} f(r) dr \right) \\
&\geq 0
\end{aligned}$$

Case 2 $\pi_k(2, r_0) =$ "fast", $\pi_k(1, r_0) =$ "slow"

$$\begin{aligned}
\Delta_{k+1}(1, r_0) = &\frac{1}{\beta + \mu_F + \lambda} \left(\underbrace{(g(2) - 2g(1) + g(0))}_{\geq 0 \text{ by Assumption 3}} + \lambda \underbrace{(J_k(3, r_0) - \bar{\mathbf{J}}_k(2) - J_k(2, r_0) + \bar{\mathbf{J}}_k(1))}_{\geq 0 \text{ per the induction hypothesis}} \right) \\
&+ \frac{1}{\beta + \mu_F + \lambda} \left(\mu_F R_F(r_0) - \mu_S R_S(r_0) - (\mu_F - \mu_S) \underbrace{(J_k(1, r_0) - J_k(0))}_{\leq \frac{\mu_F R_F(r_0) - \mu_S R_S(r_0)}{\mu_F - \mu_S}} + \mu_F \bar{\mathbf{J}}_k(1) \right) \\
&+ \frac{1}{\beta + \mu_F + \lambda} \left(- \int_0^{\hat{r}_1} (\mu_F R_F(r) + \mu_F J_k(0)) f(r) dr - \int_{\hat{r}_1}^{\infty} (\mu_S R_S(r) + \mu_S J_k(0) + (\mu_F - \mu_S) J_k(1, r)) f(r) dr \right) \\
&\geq \frac{1}{\beta + \mu_F + \lambda} \Bigg(\mu_F \bar{\mathbf{J}}_k(1) - \int_0^{\hat{r}_1} (\mu_F R_F(r) + \mu_F J_k(0)) f(r) dr \\
&\qquad - \int_{\hat{r}_1}^{\infty} (\mu_S R_S(r) + \mu_S J_k(0) + (\mu_F - \mu_S) J_k(1, r)) f(r) dr \Bigg) \\
&\geq 0 \qquad \text{by proceeding as in Case 1}
\end{aligned}$$

Case 3 $\pi_k(2, r_0) =$ "slow", $\pi_k(1, r_0) =$ "slow"

$$\Delta_{k+1}(1, r_0) = \frac{1}{\beta + \mu_F + \lambda} \left(\underbrace{(g(2) - 2g(1) + g(0))}_{\geq 0 \text{ by Assumption 3}} + \lambda \underbrace{(J_k(3, r_0) - \bar{\mathbf{J}}_k(2) - J_k(2, r_0) + \bar{\mathbf{J}}_k(1))}_{\geq 0 \text{ per the induction hypothesis}} + \underbrace{\mu_S R_S(r_0) - \mu_S R_S(r_0)}_{=0} \right)$$

$$+\frac{1}{\beta+\mu_F+\lambda}\left(\underbrace{\mu_S\bar{\mathbf{J}}_k(1)+(\mu_F-\mu_S)J_k(2,r_0)-\mu_S J_k(0)-(\mu_F-\mu_S)J_k(1,r_0)+\mu_F J_k(0)}_{=(\mu_F-\mu_S)\left(J_k(2,r_0)-\bar{\mathbf{J}}_k(1)-J_k(1,r_0)+J_k(0)\right)+\mu_F\bar{\mathbf{J}}_k(1)\geq\mu_F\bar{\mathbf{J}}_k(1)}\right)$$
$$+\frac{1}{\beta+\mu_F+\lambda}\left(-\int_0^{\hat{r}_1}\left(\mu_F R_F(r)+\mu_F J_k(0)\right)f(r)dr-\int_{\hat{r}_1}^{\infty}\left(\mu_S R_S(r)+\mu_S J_k(0)+(\mu_F-\mu_S)J_k(1,r)\right)f(r)dr\right)$$

Therefore:

$$\begin{aligned}\Delta_{k+1}(1,r_0)&\geq\frac{1}{\beta+\mu_F+\lambda}\Bigg(\mu_F\bar{\mathbf{J}}_k(1)-\int_0^{\hat{r}_1}\left(\mu_F R_F(r)+\mu_F J_k(0)\right)f(r)dr\\&\qquad-\int_{\hat{r}_1}^{\infty}\left(\mu_S R_S(r)+\mu_S J_k(0)+(\mu_F-\mu_S)J_k(1,r)\right)f(r)dr\Bigg)\\&\geq 0\qquad\text{by proceeding as in Case 1}\end{aligned}$$

This shows that, for any value of k and for any $r_0\in\Re+$, we have $J_k(i+1,r_0)-\bar{\mathbf{J}}_k(i)-J_k(i,r_0)+\bar{\mathbf{J}}_k(i-1)\geq 0$, for each $i\geq 1$, and $J_k(1,r_0)-J_k(0)\geq R_S$. By taking the limit when $k\to\infty$, we obtain that $J^*(i+1,r_0)-\bar{\mathbf{J}}^*(i)-J^*(i,r_0)+\bar{\mathbf{J}}^*(i-1)\geq 0,\forall i\geq 1,\forall r_0\in\Re+$, and $J^*(1,r_0)-J^*(0)\geq R_S(r_0),\forall r_0\in\Re+$. □

C.3 Proof of Proposition 3.4

C.3.1 Comparison of Risk Metrics:

Based on Equation (3.29), each risk metric $\xi^e(r_1)$, $\xi^e(r_2)$, $\xi^c(r_1)$ and $\xi^c(r_2)$ is expressed as follows:

$$\xi^e(r_1)=\frac{\sum_{i\geq1}\nu^e(i,r_1,F)p_F K(s_1)}{\sum_{i\geq1}\nu^e(i,r_1,F)}=p_F K(s_1)$$
$$\xi^e(r_2)=\frac{\sum_{i\geq1}\nu^e(i,r_2,F)p_F((1-\alpha)K(s_1)+\alpha K(s_2))}{\sum_{i\geq1}\nu^e(i,r_2,F)}=p_F((1-\alpha)K(s_1)+\alpha K(s_2))$$
$$\xi^c(r_1)=\frac{\sum_{i\geq1}\nu^c(i,r_1,F)p_F K(s_1)}{\sum_{i\geq1}\nu^c(i,r_1,F)}=p_F K(s_1)$$
$$\begin{aligned}\xi^c(r_2)&=\frac{\sum_{i\geq1}\nu^c(i,r_2,S)p_S(1-\alpha)K(s_1)+\sum_{i\geq1}\nu^c(i,r_2,F)p_F((1-\alpha)K(s_1)+\alpha K(s_2))}{\sum_{\phi\in\{S,F\}}\sum_{i\geq1}\nu^c(i,r_2,\phi)}\\&=p_S(1-\alpha)K(s_1)\frac{\sum_{i\geq1}\nu^c(i,r_2,S)}{\sum_{\phi\in\{S,F\}}\sum_{i\geq1}\nu^c(i,r_2,\phi)}+p_F((1-\alpha)K(s_1)+\alpha K(s_2))\frac{\sum_{i\geq1}\nu^c(i,r_2,F)}{\sum_{\phi\in\{S,F\}}\sum_{i\geq1}\nu^c(i,r_2,\phi)}\end{aligned}$$

Notice that $\xi^e(r_1)=\xi^c(r_1)$. Also note that $p_S K(s_1)(1-\alpha)<p_F(K(s_1)\alpha+K(s_2)(1-\alpha))$ because $p_S<p_F$ and $K(s_1)<K(s_2)$. Then we can bound $\xi^c(r_2)$ with

$$\begin{aligned}\xi^c(r_2)&<p_F(K(s_1)\alpha+K(s_2)(1-\alpha))\left(\frac{\sum_{i\geq1}\nu^c(i,r_2,S)}{\sum_{\phi\in\{S,F\}}\sum_{i\geq1}\nu^c(i,r_2,\phi)}+\frac{\sum_{i\geq1}\nu^c(i,r_2,F)}{\sum_{\phi\in\{S,F\}}\sum_{i\geq1}\nu^c(i,r_2,\phi)}\right)\\&=p_F(K(s_1)\alpha+K(s_2)(1-\alpha))=\xi^e(r_2)\end{aligned}$$

This concludes that $\xi^c(r_2)<\xi^e(r_2)$, and shows Equation (3.30).

C.3.2 Comparison of Queue Lengths:

Next we derive expected queue lengths $E^e[N]$ and $E^c[N]$. In order to compute these metrics we first need the probability distributions corresponding the policies. Note that aggressive policy's Markov Chain can be reduced to that of $M/M/1$ queuing system with arrival rate λ and service rate μ_F. Hence, $E^e[N] = \frac{\lambda}{\lambda+\mu_F}$. The balance equations of the conservative policy are as follows:

$$\begin{aligned}
\nu^c(0)\left(\lambda r + \lambda(1-r)(1-\alpha)\right) &= \mu_F \nu^c(1, r_1, F) + \mu_S \nu^c(1, r_2, S) \\
&\quad + \mu_F(1-r)\alpha(\nu^c(2, r_1, F) + \nu^c(2, r_2, F)) + \mu_S(1-r)\alpha\nu^c(2, r_2, S) && \text{(C.9)} \\
\nu^c(1, r_1, F)(\lambda + \mu_F) &= \nu^c(0)\lambda r + \mu_F r(\nu^c(2, r_1, F) + \nu^c(2, r_2, F)) + \mu_S r \nu^c(2, r_2, S) && \text{(C.10)} \\
\nu^c(1, r_2, S)(\lambda + \mu_S) &= \nu^c(0)\lambda(1-r)(1-\alpha) \\
&\quad + \mu_F(1-r)(1-\alpha)(\nu^c(2, r_1, F) + \nu^c(2, r_2, F)) + \mu_S(1-r)(1-\alpha)\nu^c(2, r_2, S) && \text{(C.11)} \\
\nu^c(2, r_1, F)(\lambda + \mu_F) &= \mu_F r(\nu^c(3, r_1, F) + \nu^c(3, r_2, F)) + \mu_S r \nu^c(3, r_2, S) + \lambda\nu^c(1, r_1, F) && \text{(C.12)} \\
\nu^c(2, r_2, F)(\lambda + \mu_F) &= \mu_F(1-r)(\nu^c(3, r_1, F) + \nu^c(3, r_2, F)) + \mu_S(1-r)\nu^c(3, r_2, S) && \text{(C.13)} \\
\nu^c(2, r_2, S)(\lambda + \mu_S) &= \lambda\nu^c(1, r_2, S) && \text{(C.14)} \\
\nu^c(3, r_1, F)(\lambda + \mu_F) &= \mu_F r(\nu^c(4, r_1, F) + \nu^c(4, r_2, F)) + \mu_S r \nu^c(4, r_2, S) + \lambda\nu^c(2, r_1, F) && \text{(C.15)} \\
\nu^c(3, r_2, F)(\lambda + \mu_F) &= \mu_F(1-r)(\nu^c(4, r_1, F) + \nu^c(4, r_2, F)) + \mu_S(1-r)\nu^c(4, r_2, S) + \lambda\nu^c(2, r_2, F) && \text{(C.16)} \\
\nu^c(3, r_2, S)(\lambda + \mu_S) &= \lambda\nu^c(2, r_2, S) && \text{(C.17)} \\
&\cdots
\end{aligned}$$

For expositional ease, we denote by $\overline{\nu}^c(i, F)$ the sum $\overline{\nu}^c(i, F) = \nu^c(i, r_1, F) + \nu^c(i, r_2, F)$. By examining the repeating portion of the Markov chain, note that $\nu^c(i, r_2, S)$ can be written as:

$$\nu^c(i, r_2, S) = \nu^c(1, r_2, S)\left(\frac{\lambda}{\lambda + \mu_S}\right)^{i-1}, \quad \forall i \geq 1 \tag{C.18}$$

We obtain after simplification of the non-repeating portion of the chain:

$$\mu_F \overline{\nu}^c(2, F) = \lambda\nu^c(1, r_1, F) + \frac{\lambda^2}{\lambda + \mu_S}\nu^c(1, r_2, S) \quad \text{from Equations (C.10),(C.11) and (C.9)} \tag{C.19}$$

$$\mu_F \overline{\nu}^c(3, F) = \lambda\overline{\nu}^c(2, F) + \frac{\lambda^3}{(\lambda + \mu_S)^2}\nu^c(1, r_2, S) \quad \text{from Equations (C.12),(C.13) and (C.19)} \tag{C.20}$$

$$\mu_F \overline{\nu}^c(i, F) = \lambda\overline{\nu}^c(i-1, F) + \frac{\lambda^i}{(\lambda + \mu_S)^{i-1}}\nu^c(1, r_2, S) \quad \text{by applying the same method recursively} \tag{C.21}$$

Using Equation (C.21), we obtain the following formula—which can be shown by a simple recursion:

$$\overline{\nu}^c(i, F) = \nu^c(1, r_1, F)\left(\frac{\lambda}{\mu_F}\right)^{i-1} + \nu^c(1, r_2, S)\lambda^i \sum_{j=1}^{i-1} \frac{1}{(\lambda + \mu_S)^{i-j}\mu_F^j} \tag{C.22}$$

Having obtained the expressions for each of the states $\overline{\nu}^c(i,F)$ and $\nu^c(i,r_2,S)$ in terms of $\nu^c(1,r_1,F)$, $\nu^c(1,r_2,S)$, we can now write the normalization equation in terms of these variables and $\nu^c(0)$:

$$\nu^c(0)+\nu^c(1,r_1,F)\sum_{i\geq 1}\left(\frac{\lambda}{\mu_F}\right)^{i-1}+\nu^c(1,r_2,S)\sum_{i\geq 1}\left(\frac{\lambda}{\lambda+\mu_S}\right)^{i-1}+\nu^c(1,r_2,S)\sum_{i\geq 2}\lambda^i\sum_{j=1}^{i-1}\frac{1}{(\lambda+\mu_S)^{i-j}\mu_F^j}=1 \tag{C.23}$$

We use Equations (C.9) and (C.10), in which we plug the expression of $\overline{\nu}^c(2,F)$ from Equation (C.19). This yields:

$$\begin{array}{lll} \nu^c(0)\lambda(r+(1-r)(1-\alpha)) & -\nu^c(1,r_2,S)(\mu_S+\lambda(1-r)\alpha) & -\nu^c(1,r_1,F)(\mu_F+\lambda(1-r)\alpha) \quad =0 \\ \nu^c(0)\lambda r & +\nu^c(1,r_2,S)\lambda r & -\nu^c(1,r_1,F)(\mu_F+\lambda(1-r)) \quad =0 \end{array}$$

Solving these equations, we obtain $\nu^c(0)$, $\nu^c(1,r_1,F)$ and $\nu^c(1,r_2,S)$ as:

$$\begin{aligned} \nu^c(0) &= \frac{\mu_S(\mu_F-\lambda)(\mu_S(\lambda+\mu_F)+\alpha\lambda(1-r)(\lambda+\mu_F)+\lambda r(\mu_F-\mu_S))}{(\alpha-1)\lambda^2\mu_F(r-1)(\lambda+\mu_F)+\mu_S^2\left(\mu_F^2+\lambda^2(r-1)\right)+\lambda\mu_S\left(\mu_F^2+\lambda^2(r-1)\right)} \\ \nu^c(1,r_1,F) &= \frac{\lambda\mu_S r(\lambda-\mu_F)(\lambda+\mu_S)}{(1-\alpha)\lambda^2\mu_F(r-1)(\lambda+\mu_F)+\mu_S^2\left(\mu_F^2+\lambda^2(r-1)\right)+\lambda\mu_S\left(\mu_F^2+\lambda^2(r-1)\right)} \\ \nu^c(1,r_2,S) &= \frac{(\alpha-1)\lambda\mu_S(r-1)(\lambda-\mu_F)(\lambda+\mu_F)}{(1-\alpha)\lambda^2\mu_F(r-1)(\lambda+\mu_F)+\mu_S^2\left(\mu_F^2+\lambda^2(r-1)\right)+\lambda\mu_S\left(\mu_F^2+\lambda^2(r-1)\right)} \end{aligned}$$

Then we compute $E^c[N]=\sum_{i\geq 1} i(\nu^c(i,r_2,S)+\nu^c(i,F))$. Following some algebra, we obtain:

$$E^c[N]-E^e[N]=\frac{\lambda(1-r)(\lambda+\mu_F)(\lambda+\mu_S)((1-\alpha)\lambda(\mu_F-\mu_S)+\mu_S((1-\alpha)\mu_F-\mu_S))}{\mu_S\left((1-\alpha)(1-r)\lambda^2\mu_F(\lambda+\mu_F)+\mu_S^2\left(\mu_F^2-\lambda^2(1-r)\right)+\lambda\mu_S\left(\mu_F^2-\lambda^2(1-r)\right)\right)} \tag{C.24}$$

The denominator is always positive (because $\mu_F>\lambda$); by examining the numerator, we obtain:

$$\begin{aligned} E^c[N]-E^e[N]<0 &\iff (1-\alpha)\lambda(\mu_F-\mu_S)+\mu_S((1-\alpha)\mu_F-\mu_S)<0 \\ &\iff \frac{(\lambda+\mu_S)(\mu_F-\mu_S)}{\lambda(\mu_F-\mu_S)+\mu_F\mu_S}<\alpha \end{aligned}$$

This shows Equation (3.31), and completes the proof. □

Bibliography

Ahmadi-Javid, A., Seyedi, P., and Syam, S. S. A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263, 2017.

Aksin, Z., Armony, M., and Mehrotra, V. The modern call center: A multi-disciplinary perspective on operations management research. *Production and operations management*, 16(6):665–688, 2007.

Akşin, Z., Ata, B., Emadi, S. M., and Su, C.-L. Structural estimation of callers' delay sensitivity in call centers. *Management Science*, 59(12):2727–2746, 2013.

Alanis, R., Ingolfsson, A., and Kolfal, B. A markov chain model for an ems system with repositioning. *Production and operations management*, 22(1):216–231, 2013.

Alizamir, S., De Véricourt, F., and Sun, P. Diagnostic accuracy under congestion. *Management Science*, 59 (1):157–171, 2013.

Alley, M., Biggs, M., Hariss, R., Herrmann, C., Li, M., and Perakis, G. Pricing for heterogeneous products: Analytics for ticket reselling. *Available at SSRN 3360622*, 2019.

Anand, K., Paç, M., and Veeraraghavan, S. Quality–speed conundrum: Trade-offs in customer-intensive services. *Management Science*, 57(1):40–56, 2011.

Andersson, T. and Värbrand, P. Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, 58(2):195–201, 2007.

Aouad, A., Farias, V., and Levi, R. Assortment optimization under consider-then-choose choice models working paper. *London Business School, London*, 2015.

Aringhieri, R., Bruni, M. E., Khodaparasti, S., and van Essen, J. T. Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Computers & Operations Research*, 78:349–368, 2017.

Armony, M., Shimkin, N., and Whitt, W. The impact of delay announcements in many-server queues with abandonment. *Operations Research*, 57(1):66–81, 2009.

Association, N. F. P. et al. *NFPA 1710, standard for the organization and deployment of fire suppression operations, emergency medical operations, and special operations to the public by career fire departments*. National Fire Protection Association, 2010.

Ata, B. and Shneorson, S. Dynamic control of an M/M/1 service system with adjustable arrival and service rates. *Management Science*, 52(11):1778–1791, 2006.

Atkinson, M. and Wein, L. Spatial queueing analysis of an interdiction system to protect cities from a nuclear terrorist attack. *Operations Research*, 56(1):247–254, 2008.

Bakır, N. A decision tree model for evaluating countermeasures to secure cargo at United States southwestern ports of entry. *Decision Analysis*, 5(4):230–248, 2008.

Balghiti, O. E., Elmachtoub, A. N., Grigas, P., and Tewari, A. Generalization bounds in the predict-then-optimize framework. *arXiv preprint arXiv:1905.11488*, 2019.

Bandara, D., Mayorga, M. E., and McLay, L. A. Optimal dispatching strategies for emergency vehicles to increase patient survivability. *International Journal of Operational Research*, 15(2):195–214, 2012.

Bandara, D., Mayorga, M. E., and McLay, L. A. Priority dispatching strategies for ems systems. *Journal of the Operational Research Society*, 65(4):572–587, 2014.

Barnett, A. CAPPS II: The foundation of aviation security? *Risk Analysis*, 24(4):909–916, 2004.

Başar, A., Çatay, B., and Ünlüyurt, T. A taxonomy for emergency service station location problem. *Optimization letters*, 6(6):1147–1160, 2012.

Bastani, H., Zhang, D., and Zhang, H. Applied machine learning in operations management. *Available at SSRN 3736466*, 2020.

Batt, R. J. and Terwiesch, C. Waiting patiently: An empirical study of queue abandonment in an emergency department. *Management Science*, 61(1):39–59, 2015.

Bélanger, V., Ruiz, A., and Soriano, P. Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*, 272(1):1–23, 2019.

Bernstein, F., Modaresi, S., and Sauré, D. A dynamic clustering approach to data-driven assortment personalization. *Management Science*, 65(5):2095–2115, 2019.

Bertsekas, D. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 3rd edition, 2005.

Bertsekas, D. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, 4th edition, 2012.

Bertsimas, D. and Kallus, N. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

Bertsimas, D., Delarue, A., and Martin, S. Optimizing schools' start time and bus routes. *Proceedings of the National Academy of Sciences*, 116(13):5943–5948, 2019.

Brotcorne, L., Laporte, G., and Semet, F. Ambulance location and relocation models. *European journal of operational research*, 147(3):451–463, 2003.

Byrne, J. P., Mann, N. C., Dai, M., Mason, S. A., Karanicolas, P., Rizoli, S., and Nathens, A. B. Association between emergency medical service response time and motor vehicle crash mortality in the united states. *JAMA surgery*, 154(4):286–293, 2019.

Cachon, G. P., Daniels, K. M., and Lobel, R. The role of surge pricing on a service platform with self-scheduling capacity. *Manufacturing & Service Operations Management*, 19(3):368–384, 2017.

Campello, F., Ingolfsson, A., and Shumsky, R. Queueing models of case managers. *Management Science*, 63 (3):882–900, 2016.

Carter, G. M., Chaiken, J. M., and Ignall, E. Response areas for two emergency units. *Operations Research*, 20(3):571–594, 1972.

Cavusoglu, H., Koh, B., and Raghunathan, S. An analysis of the impact of passenger profiling for transportation security. *Operations Research*, 58(5):1287–1302, 2010.

Chan, C., Yom-Tov, G., and Escobar, G. When to use speedup: An examination of service systems with returns. *Operations Research*, 62(2):462–482, 2014.

Chu, L. Y., Wan, Z., and Zhan, D. Harnessing the double-edged sword via routing: Information provision on ride-hailing platforms. *Available at SSRN 3266250*, 2018.

Debo, L., Toktay, L., and Van Wassenhove, L. Queuing for expert services. *Management Science*, 54(8): 1497–1512, 2008.

Dickerson, J. P., Sankararaman, K. A., Srinivasan, A., and Xu, P. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Dimitrakopoulos, Y., Economou, A., and Leonardos, S. Strategic customer behavior in a queueing system with alternating information structure. *arXiv preprint arXiv:1906.05628*, 2019.

Dobson, G. and Pinker, E. J. The value of sharing lead time information. *IIE Transactions*, 38(3):171–183, 2006.

Elmachtoub, A., Liang, J. C. N., and McNellis, R. Decision trees for decision-making under the predict-then-optimize framework. In *International Conference on Machine Learning*, pages 2858–2867. PMLR, 2020.

Elmachtoub, A. N. and Grigas, P. Smart "predict, then optimize". *Management Science*, 2021.

Ferreira, K. J., Lee, B. H. A., and Simchi-Levi, D. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.

Gallien, J., Mersereau, A. J., Garro, A., Mora, A. D., and Vidal, M. N. Initial shipment decisions for new products at zara. *Operations Research*, 63(2):269–286, 2015.

Gans, N., Koole, G., and Mandelbaum, A. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5(2):79–141, 2003.

Gendreau, M., Laporte, G., and Semet, F. The maximal expected coverage relocation problem for emergency vehicles. *Journal of the Operational Research Society*, 57(1):22–28, 2006.

George, J. and Harrison, J. Dynamic control of a queue with adjustable service rate. *Operations research*, 49 (5):720–731, 2001.

Gilliam, R. An application of queueing theory to airport passenger security screening. *Interfaces*, 9(4): 117–123, 1979.

Glaeser, C. K., Fisher, M., and Su, X. Optimal retail location: Empirical methodology and application to practice: Finalist–2017 m&som practice-based research competition. *Manufacturing & Service Operations Management*, 21(1):86–102, 2019.

Güneş, E. D., Melo, T., and Nickel, S. Location problems in healthcare. In *Location science*, pages 657–686. Springer, 2019.

Guo, P. and Zipkin, P. Analysis and comparison of queues with different levels of delay information. *Management Science*, 53(6):962–970, 2007.

Hassin, R. *Rational queueing*. CRC press, 2016.

Hassin, R. and Haviv, M. *To queue or not to queue: Equilibrium behavior in queueing systems*, volume 59. Springer Science & Business Media, 2003.

Hopp, W., Iravani, S., and Liu, F. White collar workforce management: An operations-oriented survey. *Production and Operations Management*, pages 1–32, 2007a.

Hopp, W., Iravani, S. M., and Yuen, G. Operations systems with discretionary task completion. *Management Science*, 53(1):61–77, 2007b.

Hu, M. and Zhou, Y. Price, wage and fixed commission in on-demand matching. *Available at SSRN 2949513*, 2019.

Hu, M., Li, Y., and Wang, J. Efficient ignorance: Information heterogeneity in a queue. *Management Science*, 64(6):2650–2671, 2018.

Ibrahim, R. Sharing delay information in service systems: A literature survey. *Queueing Systems*, 89(1-2): 49–79, 2018.

Ibrahim, R., Armony, M., and Bassamboo, A. Does the past predict the future? The case of delay announcements in service systems. *Management Science*, 63(6):1762–1780, 2017.

Jagtenberg, C. J., Bhulai, S., and van der Mei, R. D. Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health care management science*, 20(4):517–531, 2017a.

Jagtenberg, C. J., van den Berg, P. L., and van der Mei, R. D. Benchmarking online dispatch algorithms for emergency medical services. *European Journal of Operational Research*, 258(2):715–725, 2017b.

Lamb, M., Anderson, D., Gloria, N., Vanslow, W., and Ieraci, R. Performance audit for department of public safety bureau of emergency medical services, 2014.

Larsen, M. P., Eisenberg, M. S., Cummins, R. O., and Hallstrom, A. P. Predicting survival from out-of-hospital cardiac arrest: a graphic model. *Annals of emergency medicine*, 22(11):1652–1658, 1993.

Lee, A. and Jacobson, S. Addressing passenger risk uncertainty for aviation security screening. *Transportation science*, 46(2):189–203, 2012.

Li, X., Zhao, Z., Zhu, X., and Wyatt, T. Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, 74(3):281–310, 2011.

Lingenbrink, D. and Iyer, K. Optimal signaling mechanisms in unobservable queues. *Operations research*, 67 (5):1397–1416, 2019.

Liu, S., He, L., and Max Shen, Z.-J. On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science*, 2020.

Lyu, G., Cheung, W. C., Teo, C.-P., and Wang, H. Multi-objective online ride-matching. *Available at SSRN 3356823*, 2019.

Mandi, J., Stuckey, P. J., Guns, T., et al. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1603–1610, 2020.

Maxwell, M. S., Restrepo, M., Henderson, S. G., and Topaloglu, H. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing*, 22(2):266–281, 2010.

McLay, L., Jacobson, S., and Kobza, J. Integer programming models and analysis for a multilevel passenger screening problem. *IIE Transactions*, 39(1):73–81, 2007.

McLay, L., Lee, A., and Jacobson, S. Risk-based policies for airport security checkpoint screening. *Transportation science*, 44(3):333–349, 2010.

McLay, L. A. and Mayorga, M. E. A dispatching model for server-to-customer systems that balances efficiency and equity. *Manufacturing & Service Operations Management*, 15(2):205–220, 2013a.

McLay, L. A. and Mayorga, M. E. A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions*, 45(1):1–24, 2013b.

Naor, P. The regulation of queue size by levying tolls. *Econometrica*, pages 15–24, 1969.

Nasrollahzadeh, A. A., Khademi, A., and Mayorga, M. E. Real-time ambulance dispatching and relocation. *Manufacturing & Service Operations Management*, 20(3):467–480, 2018.

Nikolaev, A., Jacobson, S., and McLay, L. A sequential stochastic security system design problem for aviation security. *Transportation Science*, 41(2):182–194, 2007.

Özkan, E. and Ward, A. R. Dynamic matching for real-time ride sharing. *Stochastic Systems*, 2020.

Paul, K. The new system Uber is implementing at airports has some drivers worried, 2015. [accessed on January 29, 2020].

Powell, W. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, 2nd edition, 2007.

Romanyuk, G. Ignorance is strength: Improving the performance of matching markets by limiting information. In *Harvard Univeristy Cambridge, MA Working Paper*. 2017.

Romanyuk, G. and Smolin, A. Cream skimming and information design in matching markets. *American Economic Journal: Microeconomics*, 11(2):250–76, 2019.

Rosenblat, A. and Stark, L. Algorithmic labor and information asymmetries: A case study of Uber's drivers. *International Journal of Communication*, 10:27, 2016.

Sampalis, J. S., Denis, R., Lavoie, A., Frechette, P., Boukas, S., Nikolis, A., Benoit, D., Fleiszer, D., Brown, R., Churchill-Smith, M., et al. Trauma care regionalization: a process-outcome evaluation. *Journal of Trauma and Acute Care Surgery*, 46(4):565–581, 1999.

Saydam, C., Rajagopalan, H. K., Sharer, E., and Lawrimore-Belanger, K. The dynamic redeployment coverage location model. *Health Systems*, 2(2):103–119, 2013.

Schmid, V. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European journal of operational research*, 219(3):611–621, 2012.

Sinha, A., Kar, D., and Tambe, M. Learning adversary behavior in security games: A pac model perspective. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 214–222. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

Sinha, A., Fang, F., An, B., Kiekintveld, C., and Tambe, M. Stackelberg security games: Looking beyond a decade of success. In *IJCAI*, pages 5494–5501, 2018.

Smith, D. R. and Whitt, W. Resource sharing for efficiency in traffic systems. *Bell System Technical Journal*, 60(1):39–55, 1981.

Stidham, S. and Weber, R. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations research*, 37(4):611–625, 1989.

Sudtachat, K., Mayorga, M. E., and McLay, L. A. Recommendations for dispatching emergency vehicles under multitiered response via simulation. *International Transactions in Operational Research*, 21(4): 581–617, 2014.

Sudtachat, K., Mayorga, M. E., and Mclay, L. A. A nested-compliance table policy for emergency medical service systems under relocation. *Omega*, 58:154–168, 2016.

Taylor, T. A. On-demand service platforms. *Manufacturing & Service Operations Management*, 20(4):704–720, 2018.

UberPeople.net. Passenger app no longer shows cars in area. https://uberpeople.net/threads/passenger-app-no-longer-shows-cars-in-area.136892/, 2017. [Online; accessed 17-April-2020].

US Customs and Border Protection. On a Typical Day in Fiscal Year 2016. Available at https://www.cbp.gov/newsroom/stats/typical-day-fy2016, 2016.

Van Barneveld, T., van der Mei, R. D., and Bhulai, S. Compliance tables for an ems system with two types of medical response units. *Computers & Operations Research*, 80:68–81, 2017.

Van Barneveld, T. The minimum expected penalty relocation problem for the computation of compliance tables for ambulance vehicles. *INFORMS Journal on Computing*, 28(2):370–384, 2016.

Van Mieghem, J. A. Dynamic scheduling with convex delay costs: The generalized c| mu rule. *The Annals of Applied Probability*, pages 809–833, 1995.

Wang, H. and Yang, H. Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological*, 129:122–155, 2019.

Wang, K. and Jacquillat, A. From classification to optimization: A scenario-based robust optimization approach. *Available at SSRN 3734002*, 2020.

Wang, X., Debo, L., Scheller-Wolf, A., and Smith, S. Design and analysis of diagnostic service centers. *Management Science*, 56(11):1873–1890, 2010.

Wein, L., Wilkins, A., Baveja, M., and Flynn, S. Preventing the importation of illicit nuclear materials in shipping containers. *Risk Analysis*, 26(5):1377–1393, 2006.

Weinberger, L. M., Gibbon, M. H., and Gibbon, J. H. Temporary arrest of the circulation to the central nervous system: I. physiologic effects. *Archives of Neurology & Psychiatry*, 43(4):615–634, 1940.

Wrzaczek, S., Kaplan, E., Caulkins, J., Seidl, A., and Feichtinger, G. Differential terror queue games. *Dynamic Games and Applications*, 7(4):578–593, 2017.

Yu, Q., Allon, G., and Bassamboo, A. How do delay announcements shape customer behavior? an empirical study. *Management Science*, 63(1):1–20, 2017.

Zhan, D. and Ward, A. Threshold routing to trade off waiting and call resolution in call centers. *Manufacturing & Service Operations Management*, 16(2):220–237, 2013.

Zhang, Z. Performance analysis of a queue with congestion-based staffing policy. *Management Science*, 55(2): 240–251, 2009.

Zhang, Z., Luh, H., and Wang, C.-H. Modeling security-check queues. *Management Science*, 57(11):1979–1995, 2011.