THE UNIVERSITY OF CHICAGO

# DESIGNING SERVICE MENUS FOR BIPARTITE QUEUEING SYSTEMS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE UNIVERSITY OF CHICAGO
BOOTH SCHOOL OF BUSINESS
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

BY
LISA CATHERINE AOKI HILLAS

CHICAGO, ILLINOIS
AUGUST 2023

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

I would like to begin by thanking my advisers René Caldentey and Varun Gupta for their support and guidance throughout my PhD. I would also like to thank Chris Ryan, who helped me begin my research journey at Booth. I could not have asked for kinder mentors.

I am grateful to Philipp Afèche, Amy Ward, and Yuan Zhong for serving on my committee. I appreciate the time they committed to the task, as well as their helpful suggestions and feedback.

I would like to thank the Booth PhD Office for their patience and support throughout my PhD. I am particularly grateful to Cynthia Hillman for her guidance while I was on the job market, and for the many helpful programs she organised.

I would not have finished this PhD without the support of my friends. I am grateful to my cohort, Cagla, Zuguang, Amir, Monty, and Alex, and others in the Operations Management group with me including Charlie, Gorkem and Yueyang. I also thank my non Operations Management friends in Chicago, in particular Lily, Francesca, and Christoph. Your friendship has made me grateful I chose Chicago to do my PhD at. And I thank my New Zealand friends, Hayden and Alice. Alice, you've been an amazing work bud, and I think by rights half of this PhD is yours.

Finally, I would like to thank my family. You have made me the person I am (so you can only blame yourselves), and that person apparently has a PhD now. And I would not be me if I didn't thank the four-legged members of my family, Rosa, Ragamuffin, Harley, and Paddy.

# ABSTRACT

This dissertation examines the performance analysis and design of multi-class multi-server bipartite queueing systems under a FCFS-ALIS service discipline. The class of queueing systems we look at have $m$ servers with exponentially distributed service times organised into $n$ service classes, where a service class is defined by the subset of servers that it is compatible with. We begin by analysing the performance of the system with fixed arrival rates into the different service classes under conventional heavy-traffic conditions, where the traffic intensity approaches one from below. Building upon the formulation and results of Afèche et al. (2021), we generalize the model by allowing the vector of arrival rates to approach the heavy-traffic limit from an arbitrary direction. We characterize the steady-state waiting times of the various service classes and demonstrate that a much wider range of waiting time outcomes is achievable when the direction of approach is generalised. Furthermore, we establish that the matching probabilities, i.e., the probabilities of different customers who join different service classes being served by different servers, do not depend on the direction along which the system approaches heavy traffic. We also investigate the design of compatibility between service classes and servers, finding that a service provider who has complete control over the matching can design a delay-minimizing menu by considering only the limiting arrival rates. When some constraints on the compatibility structure exist, the direction of convergence to heavy-traffic affects which menu minimizes delay. Additionally, we discover that the bipartite matching queueing system exhibits a form of Braess's paradox, where adding more connectivity to an existing system can lead to higher average waiting times, even when neither customers nor servers are acting strategically.

We then extend the model to allow for strategic behaviour. We assume that customers of different types have heterogeneous preferences over the many servers available. The goal of the service provider is to design a menu of service classes that balances two competing objectives: (1) maximize customers' average matching reward and (2) minimize customers' av-

erage waiting time. Customers act as rational self-interested utility maximizing agents when choosing which service class to join. In particular, they join the class that maximizes their expected ex-ante net utility, which is given by the difference between the server-dependent service reward they receive minus a disutility based on the mean steady-state waiting time of the service class they join. We study the menu design problem under conventional heavy traffic conditions. For the case of two servers, we provide a complete characterization of the possible menus and their delay-reward tradeoffs. For general number of servers, we prove that if the service provider only cares about minimizing average delay or maximizing total matching reward then very simple menus are optimal. Finally, we provide Mixed Integer Linear Programming (MILP) formulations for optimizing the delay-reward trade-off within fairly rich and practically relevant families of menus, which we term *Partitioned* and *Tailored*.

# CHAPTER 1

# INTRODUCTION

Multi-class multi-server queueing systems are used to model many real-world settings, including applications such as public housing, health care, the adoption of children, and manufacturing. These settings can experience extremely high levels of congestion. For example, the Chicago Housing Authority reported more than 170,000 families waiting for public housing in 2021 (Sheridan (2022)). Similarly, in the same year, about 113,589 children in the United States were waiting to be adopted (Duffin (2022)). In the healthcare system, more than 100,000 people are waiting for an organ transplant at any given moment in time, with average waiting times that can be as long as 5 years for a kidney transplant according to the National Kidney Foundation. Because of these high levels of congestion, any improvements we can make to the design of these systems can have significant benefits for the people waiting.

This dissertation contributes to the literature on the analysis and design of multi-class multi-server queueing systems. The particular type of multi-class multi-server queueing systems we consider are systems in which $m$ heterogeneous servers are organised into $n$ service classes, where each service class is compatible with a particular subset of servers. We assume a first-come-first-served assign-longest-idle-server (FCFS-ALIS) service discipline is used. That is, when a server finishes serving a customer, they consider all of the customers that belong to classes they are compatible with, and serve the customer that has been waiting the longest. Similarly, if a customer were to arrive to the system at a time in which multiple servers she is compatible with are idle and available to serve her, then she would be served by the server that had been idle the longest. While the FCFS-ALIS assumption is restrictive, and may have negative implications in terms of performance, it is an important service discipline to study as it is simple and easy to implement, and widely used in practice.

It also has an appealing notion of fairness, which is important in some of our motivating examples such as public housing. In addition to the FCFS-ALIS assumption, we also assume that customers arrive to the system according to independent Poisson processes, and servers have exponential service rates which depend only on the server.

While these models are useful for studying many different applications, they can be both analytically and computationally intractable, making questions of performance analysis and system design difficult to answer. Because of this, we study the problem under conventional heavy-traffic conditions, in which we consider a sequence of systems where the service rates and the menu of service classes remain fixed, and we increase the arrival rates into the different service classes until the sum of the arrival rates is equal to the total service capacity. We then calculate the limiting outcomes of this sequence of systems. By using a heavy-traffic scaling, we are able to provide approximations of these systems that are much simpler to analyse and reveal fundamental properties of the system. The heavy-traffic assumption is not only mathematically more tractable, it is also very appropriate for our motivating examples, which in the real world are operating with very high levels of congestion.

In Chapter 2, which is based on Hillas et al. (2023), we consider the problem when the arrival rates into the different service classes are fixed. We study the problem under conventional heavy-traffic conditions, in which we consider a sequence of systems where the service rates and the menu of service classes remain fixed, and we increase the arrival rates into the different service classes until the sum of the arrival rates is equal to the total service capacity. We develop heavy-traffic machinery to calculate the limiting expected delays customers face, and the expected matching probabilities, that is, the probabilities with which different customers are served by different servers at the limit of this sequence of systems. In doing so, we are generalising the results of Afèche et al. (2021), by using a more general heavy-traffic scaling. This generalisation is important as it allows for a wider range of waiting time outcomes to be observed for the same limiting vector of arrival rates.

To see this, consider a system with two independent M/M/1 queues, both being served at rate 1. Using a conventional heavy traffic scaling, in which the number of servers and the service rates remain fixed, and the traffic intensity approaches 1 from below, the limiting arrival rates of both queues will be 1. The heavy traffic scaling in Afèche et al. (2021) has the proportion of customers arriving into the different queues remaining constant while taking the limit. However, if we do this in our simple M/M/1 example, we can see that this would limit us to concluding that the heavy traffic delays of both queues are equal. If instead we generalise the approach to heavy-traffic, allowing the arrival rates into the different queues to approach their limits at different rates, we are able to choose parameters such that the queues will experience different heavy-traffic delays. We can interpret the different rates of approach in the real world as the different queues having arrival rates closer or further away to their predicted limiting value.

This allows for a wider range of scenarios to be modelled accurately, which is necessary in Chapter 3 in order to extend the model to allow for customers to strategically choose which service classes to join. This extension also motivates us to allow for queues with no arrivals. This can be important for developing a coherent model when including strategic behaviour. In this case, it is possible to offer queues that no customers will choose to join, but we still need to calculate expected delays for those queues in order to justify why customers are not choosing to join them.

We begin Chapter 2 by calculating the waiting time and matching probability outcomes in heavy-traffic. We show that different approaches to the heavy-traffic limit produce different limiting waiting times, while the matching probabilities depend only on the limiting vector of arrival rates. Based on this, we demonstrate through an example that very minor perturbations in arrival rates can produce significant improvements in waiting time outcomes in the pre-limit. We end Chapter 2 by discussing some basic questions regarding the design of menus of service classes when customers are not behaving strategically and arrival rates

into the different service classes are fixed. We find that when the service provider has complete control over the compatibility structure, they only need to consider the limiting arrival rates in order to design a delay minimising menu. When there are some constraints on the compatibility structure, then the particular approach to heavy-traffic does affect which menu minimises delay.

In Chapter 3, which is based on Hillas et al. (2023), we use the machinery developed in Chapter 2 to study the question of how to design multi-class multi-server queueing systems when customers are allowed to strategically choose for themselves which queues to join. In particular, we consider the problem of designing a matching queueing system such as the one depicted in Figure 1.1. As in Chapter 2, servers are heterogeneous in terms of the amount of time it takes them to serve a customer (i.e., have different service rates $\mu_j$), and are organised into a collection of $n$ service classes. What is different in Chapter 3 is that instead of the arrival rates into the different service classes being exogenously given, customers of different types $\theta = 1, \ldots, \Theta$ arrive to the system at rates $\alpha_\theta$ and upon arrival are able to choose for themsleves which service classes to join. Additionally, we now assume that not

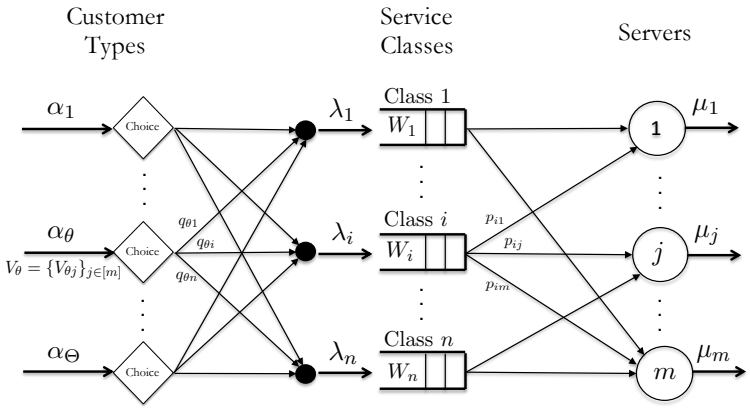



Figure 1.1: A multi-class multi-server matching queueing system.

only are servers heterogeneous in their arrival rates, they are heterogenous in other attributes that affect the reward $\{V_{\theta j}\}$ that customers receive for the service. Customers choose which

service classes to join by trading off the expected value of service they will receive from each service class against the expected delay at the service classes.

The goal of the service provider is to design a service mechanism that will match customers to servers and will balance two (usually) competing objectives: (1) maximize customers' average matching service reward and (2) minimize customers' average waiting time. We will restrict ourselves to a special class of mechanisms in which the service provider offers a static menu of service classes $i = 1, \ldots, n$ and customers choose which one of them to join upon arrival. As in Chapter 2, a service class is defined by the subset of servers that are able to serve customers who join that class.

We begin by defining a strategic equilibrium in the pre-limit, where we use a Nash equilibrium type concept. We explore some of the features and challenges of this model when there are only two servers. We find that when there are only two servers, there are significant simplifications to the analysis we can make relative to the general case. However, even with these simplifications, equilibrium analysis and the menu design problem cannot be completely studied analytically, but must be studied computationally.

In order to tackle the problem more generally, we extend our notion of a strategic equilibrium to heavy-traffic. We define and prove the existence of heavy-traffic equilibria in this setting. We explore menus that will be optimal for a service provider looking to maximise the value achieved by serving customers using servers they most prefer, and menus that will be optimal for a service provider looking to minimise delay. In some cases it is possible to achieve both of these goals simultansouly, and we provide necessary and sufficient conditions to understand when this is possible. For when this is not possible, we provide mixed-integer linear-programmes (MILPS) that identify menus that trade-off the performance of these two objectives. We complete the chapter with a simple numerical analysis to evaluate the performance of our MILPS.

## 1.1 Related Literature

**Related Literature.** Heavy-traffic approximations have long been used to simplify the study of intractable queueing systems. Early works in this area include Kingman (1962) and Whitt (1974). These papers look at a so-called "conventional" approach to heavy-traffic, in which the number of servers and their service capacities remain fixed, and the arrival rate grows large in such a way that the traffic intensity of the system converges to one from below. An alternative class of "many-server" heavy-traffic limits have also been considered in the literature by carefully letting the number of servers and arrival rate grow unboundedly, e.g., Halfin and Whitt (1981) or Atar (2012). Motivated by mathematical tractability as well as by the fact that many real-world service systems operate under high levels of congestion, we will study the performance of our multi-class multi-server bipartite queuing system operating under conventional heavy traffic conditions.

A range of questions can be answered using heavy-traffic approximations. In the context of parallel service systems, Harrison and Lopez (1999) study the question of optimal control of parallel service systems, that is, which servers should be used to serve which service classes, and in which order should the different service classes be served. Harrison and Lopez (1999) solve an approximating Brownian control problem, and conjecture that a discrete review policy will minimise holding costs for the original queuing system. This approach of using an approximating Brownian control problem to develop an optimal policy was originally suggested by Harrison (1988). Williams (2000) and Bell and Williams (2001) go on to prove the asymptotic optimality of a continuous review policy for a two-server system. Following this work, Mandelbaum and Stolyar (2004a) proves the asymptotic optimality of the $c\mu-$rule for convex holding costs. A distinctive feature in all of these papers is that they impose a *complete resource pooling* condition on the connectivity and/or compatibility between service classes and servers (see Harrison and Lopez, 1999). Roughly speaking, this condition boils

down to assuming that the servers' capacities can be pooled together so that the servers can essentially act as a single "super-server". This assumption significantly simplifies the analysis as it allows us to obtain a single-dimensional state-space description of the workload of the system in the heavy traffic limit.

The complete resource pooling assumption is quite restrictive, however, and can be shown not to hold when strategic customer behaviour is allowed as in Caldentey et al. (2023). There has already been some work moving beyond the complete resource pooling assumption. Kushner and Chen (2000) prove the convergence to the heavy-traffic limit of a particular class of systems that do not satisfy the complete resource pooling assumption under quite general conditions. Pesic and Williams (2016) generalises Harrison and Lopez (1999) beyond the complete resource pooling assumption. Other works analysing multi-class multi-server queueing systems with no complete resource pooling assumption include Shah and de Veciana (2016) and Hurtado Lange and Maguluri (2022). Shah and de Veciana (2016) look at a system in which servers simultaneously work to process the same job, while Hurtado Lange and Maguluri (2022) analyse a generalised switch problem under a MaxWeight service policy.

In addition to studying the problem of optimal control, questions regarding the performance of parallel service systems have been studied using heavy-traffic approximations, or fluid approximations more generally. Talreja and Whitt (2008) looks at the problem of calculating matching rates for a parallel service system operating under FCFS, that is, with what probability is each service class served by each server, although the authors looked at this question for an overloaded system with abandonments. Matching rates were calculated for specific classes of networks. Various approximation methods have been developed for calculating matching rates including the *dissipative* algorithm proposed by Caldentey and Kaplan (2002), a related approximation based on Ohm's law proposed by Fazel-Zarandi and Kaplan (2018) and a quadratic programming formulation proposed by Afèche et al. (2021). Of these papers looking at the performance of parallel service systems under FCFS, Afèche

et al. (2021) is the only one to also look at calculating waiting times as we do here. Another contribution of Afèche et al. (2021) is to study the question of the design of matching topologies fixing the scheduling policy. While Afèche et al. (2021) studies this design question for a FCFS service discipline, Varma and Maguluri (2021) studies the same question of the design of matching topologies under a MaxWeight service discipline.

The specific model we look at here is a generalisation of Afèche et al. (2021), which itself developed out of a long history of papers studying bipartite queueing systems and bipartite matching models under an FCFS service discipline. Early papers in this area include Schwartz (2004) and Green (1985), who look at the steady-state performance of these systems given a particular hierarchical compatibility structure between service classes and service classes, and Kaplan (1984, 1988), who similarly analysed the steady-state performance of parallel queuing systems, but for more general compatibility structures. Following Kaplan (1984, 1988), Kaplan's multi-class multi-server queueing model was adapted by Caldentey and Kaplan (2002), who introduced an infinite-bipartite matching model to analyse matching probabilities under a FCFS service discipline. The infinite matching model was further developed in Caldentey et al. (2009), Bušić et al. (2013), Adan and Weiss (2012), Adan et al. (2018a) and Fazel-Zarandi and Kaplan (2018). The connection between the steady-state solution of the queueing model and the infinite bipartite matching model was formalized by Adan and Weiss (2014) under the FCFS-ALIS service discipline (see also Adan et al., 2018b, 2019 and the survey by Gardner and Righter, 2020).

Since the development of the infinite matching model and the queueing model, different authors have looked at different aspects of the problem. Bušić et al. (2013), Mairesse and Moyal (2017), and Moyal and Perry (2017) look at stability conditions of such systems, and find that the system will be stable so long as a set of Hall's type conditions are satisfied. Also of interest are the steady-state matching probabilities. Caldentey et al. (2009) were able to use a particular Markov chain representation to calculate the steady-state distribu-

tion of the matching system for particular classes of matching topologies. Adan and Weiss (2012) came up with an alternative Markov chain representation to derive the steady-state distribution of the matching system for general matching topologies, while Adan and Weiss (2014) used a similar approach to look at the multi-class multi-server queueing problem, and showed the equivalence of the steady-state outcomes for the matching and the overloaded queueing system. However, the combinatorial structure of the state space description of the Markov chain limits the size of the systems that can be studied both analytically and computationally. Afèche et al. (2021) use heavy traffic analysis to unveil a number of structural properties embedded in the infinite matching model and its corresponding multi-class bipartite matching queueing system (see also the survey by Gardner and Righter, 2020 for a comprehensive review of related papers and models).

For the most part, the aforementioned stream of literature has assumed that the *matching topology* connecting services classes to servers is exogenously given and has focused on the performance analysis of the queueing system; i.e., identifying conditions that ensure stability or characterizing steady-state matching rates. The problem of designing optimal matching topologies is studied in Afèche et al. (2021) under the assumption that consumers are passive agents who do not choose which service class to join. In this setting, they can restrict themselves to topologies in which there is a one-to-one correspondence between customer types and service classes and so the design problem reduces to deciding the subset of servers that should serve each service class. To deal with the combinatorial complexity of the problem identified by Adan and Weiss (2014), Afèche et al. (2021) rely on a heavy traffic analysis that unveils a surprisingly simple structure. Namely, under heavy-traffic conditions, they show that any bipartite matching system can be partitioned into a collection of complete resource pooling (CRP) subsystems, which are interconnected using a direct acyclic graph (DAG). The significance of these results is that they allow us to replace the combinatorial structure of the original queueing system (expressed in terms of permutations of servers)

with a more aggregate representation defined by the collection of topological orders of the CRP components. As a result, they show that the DAG together with the aggregate service capacity on each CRP component fully determines the vector of steady-state waiting times. Combining this insight with a Quadratic Programming approach to approximate matching flows, Afèche et al. (2021) propose a mixed-integer linear program formulation that can be used to characterize the set of matching topologies that optimize the tradeoff between matching rewards and waiting times in a Pareto efficiency sense.

This work builds on and extends Afèche et al. (2021) first by considering a more general heavy-traffic scaling, and also by allowing consumers to choose the service class they want to join. In Chapter 2, we will see that our more general heavy-traffic scaling allows for a wider range of waiting time outcomes to be modelled. In Chapter 3, we will see that the extension to strategic customers is not trivial. For one, the number of service classes can no longer be reduced to the number of customer types as the service provider can in principle offer a full service menu with as many service classes as the number of possible subsets of the servers. Also, by allowing customers to self-select the service class they want to join, the service provider has less control over the final matching. In other words, while in Afèche et al. (2021) the service provider acts as a *central planner* that has full control over how to route customers to service classes, in our case the central planner acts as a *principal* that can only induce *agents* (customers) to join a particular service class by designing an *incentive compatible* menu. The Principal-Agent nature of our problem implies that waiting times and matching flows must be computed while imposing equilibrium conditions, which brings an extra layer of complexity to the problem. Finally, another subtle but important difference between Afèche et al. (2021) and our paper relates to how a heavy traffic analysis can be conducted. Specifically, in Afèche et al. (2021) the heavy traffic limit was essentially exogenously defined by letting the vector of customers' arrival rates converge (from below) along a pre-specified direction to a limiting vector of arrival rates. In contrast, in our case in

which customers self-select the service class they want to join, the direction of convergence to heavy traffic is endogenously determined in equilibrium.

A distinctive feature of many of the papers that we have discussed so far, and which is also central to our work, is the FCFS-ALIS service discipline that is used in the matching of customers and servers. This type of service discipline is appropriate in settings (such as public housing allocations, adoption agencies or state-run nursing homes, to name a few) in which fairness considerations and/or legal regulations prevent the service provider from implementing other types of priority-based policies that could be (or could be perceived to be) discriminatory. If we relax this requirement, there exists a vast queueing literature on skill-based routing devoted to the problem of characterizing dynamic scheduling policies to control and optimize the flow of customers in a multi-server setting. Some representative examples of this stream of work include Harrison (1998), Harrison and Lopez (1999), Mandelbaum and Stolyar (2004b), Atar (2005), Bell and Williams (2005), Wallace and Whitt (2005), Dai and Tezcan (2005), Gurvich and Whitt (2009, 2010), Ward and Armony (2013), and Comte (2019). Comte (2019) in particular like us consider the routing from arrivals to service classes and service classes to servers separately, although their model does not include strategic behaviour.

Another stream of papers that is relevant to our work is concerned with the design of differentiated service menus. Some representative papers in this area include Van Mieghem (2000), Plambeck (2004), Maglaras and Zeevi (2005), Afèche (2013), Afèche and Pavlin (2016), Nazerzadeh and Randhawa (2018), Afèche et al. (2021), Ashlagi et al. (2021) and Ashlagi et al. (2022). The typical setting in these papers is one in which customers are heterogeneous in terms of their valuation or willingness-to-pay for service and their sensitivity to delay, while servers are homogeneous (in many cases a single server is considered). Under these conditions, a service class consists of two components: (1) the price that the service provider charges for the service and (2) the expected waiting time. Operationally, the service

provider controls the service discipline which allows her to offer differentiated waiting times to the different service classes. The goal of the service provider is to design a menu of service classes that maximizes her profit or in some cases a social welfare objective.

In terms of applications, stochastic matching systems have been extensively used in the healthcare literature to study organ transplantation (e.g., Zenios et al. 2000, Akan et al. 2012, Bertsimas et al. 2013 and Ding et al. 2018) and kidney exchanges (e.g., Unver (2010), Anderson et al. 2017, Ashlagi et al. 2018b and Akbarpour et al. 2018). Other applications include public housing (e.g., Bloch and Cantala 2017, Leshno 2017, and Arnosti and Shi 2018), adoptions (e.g., Baccara et al. 2014 and Slaugh et al. 2016), labor markets (e.g., Rogerson et al. 2005, Arnosti et al. 2018 and Baccara et al. 2018), assemble-to-order manufacturing (e.g., Gurvich and Ward, 2014 and Nazari and Stolyar, 2016) and process flexibility (e.g., Jordan and Graves, 1995, Bassamboo et al., 2012, Tsitsiklis and Xu, 2012, 2017 and Shi et al., 2018).

## 1.2 Notation

To simplify notation, we will adopt the following conventions. For a positive integer $k \in \mathbb{N}$, we let $[k] := \{1, 2, \ldots, k\}$. For a vector $x = (x_i)_{i\in[k]}$ and a subset $S \subseteq [k]$, we let $x_S := \sum_{i\in S} x_i$ and $|x| := x_{[k]} = \sum_{i=1}^{k} x_i$. All vectors are column vectors, and for a vector $x \in \mathbb{R}^k$, we let $|x| := \sum_{i\in[k]} x_i$.

# CHAPTER 2

# HEAVY TRAFFIC ANALYSIS OF BIPARTITE QUEUEING SYSTEMS

## 2.1 Introduction

In this chapter, we anayse the performance of multi-class bipartite queueing systems for fixed matching topologies under an FCFS-ALIS service discipline. Multi-class bipartite queuing systems are used for modelling a variety of important applications, such as public housing, health-care, and manufacturing. However, these models can be both analytically and computationally intractable, making questions of performance analysis and system design difficult to answer. Because of this, we use a heavy-traffic scaling to provide approximations of these systems that are much simpler to analyse and reveal fundamental properties of the system.

The specific model in this chapter has $n$ service classes and $m$ distinct servers. Customers arrive to each class according to independent Poisson processes. Service times are exponentially distributed, with service rates depending only on the server, and not on the service class. Each service class has a particular subset of servers they can be served by. Each server may potentially be compatible with multiple service classes. Servers serve the service classes they are compatible with according to a FCFS-ALIS service discipline. That is, when a server finishes serving a customer, they consider all of the customers that belong to classes they are compatible with, and serve the customer that has been waiting the longest. Similarly, if a customer were to arrive to a service class and find multiple servers they are compatible with idle, they would be assigned to the server that had been idle the longest.

We analyse two aspects of the performance of this model, the expected waiting times of the different service classes, and the matching probabilities of the different service classes, that is, the probability with which a customer of a given class is served by a particular server.

The chapter is organized as follows. In Section 2.2 we provide a detailed mathematical description of the bipartite queueing model, review some related results in the literature and introduce the heavy traffic regime that we will use to analyze the performance of the system. Section 2.3 is devoted to the derivation of the limiting steady-state waiting times of the different service classes. Our main result in this section is Theorem 1 which provides a complete characterization of these limiting waiting times in terms of an underlying set of complete resource pooling components and their connectivity that emerge under heavy traffic. In Section 2.4 we study the steady-state matching probabilities between service classes and servers and show in Theorem 2 that these probabilities do not depend on the particular direction along which the system reaches heavy traffic. This is in direct contrast to the behaviour of the steady-state waiting times, which are particularly sensitive to the direction of convergence. In Section 2.5 we discuss a number of insights that emerge from our theoretical results. For instance, what vectors of delays are implementable and how to design the connectivity between service classes and servers to achieve them. We also show that adding more connectivity to an existing bipartite queueing system can lead to longer average delays (i.e., some form of Braess's paradox). Section 2.6 contains the proofs and additional discussion of our main results Theorems 1 and 2. Finally, the Appendix contains additional proofs of various intermediate results. Concluding remarks and discussion of future directions can be found in Chapter 4

## 2.2 Model Description

In this section, we provide a detailed mathematical description of the model and basic definitions.

We consider a service system as follows. We have a set of $m$ servers organised into a set of $n$ service classes. Each service class is served by a particular subset of servers. This

information is encoded in a compatibility matrix $M \in \{0,1\}^{n\times m}$, where service class $i$ can be served by server $j$ iff $m_{ij} = 1$. Customers arrive to the service classes according to independent Poisson processes. We let $\lambda = (\lambda_1, ..., \lambda_n)$ be the arrival rates into the different service classes. Service times are exponentially distributed, and depend only on the server. The vector of service rates will be denoted by $\mu = (\mu_1, ..., \mu_m)$. Servers will serve customers they are compatible with according to a FCFS-ALIS service discipline.

To illustrate, Figure 2.1 depicts an example with four servers ($m = 4$), and four service classes ($n = 4$).

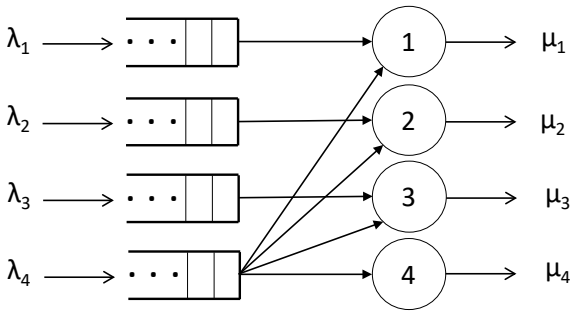



Figure 2.1: Example with four service classes and four servers.

In this example, the menu $M$ is given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{2.1}$$

that is, class 1 is compatible with server 1; class 2 is compatible with server 2; class 3 is compatible with server 3; and class 4 is compatible with all servers. Note that a server may belong to multiple service classes.

We are only interested in systems which operate with stable queue lengths. The following

result, from Adan and Weiss (2014) tells us exactly which triplets $(\lambda, \mu, M)$ produce stable steady-state outcomes.

**Proposition 1.** (Adan and Weiss, 2014, Theorem 2.1) *For a menu $M$ with arrival rates $\lambda$ and service rates $\mu$, define the slack of a set of servers $\Delta_{\mathscr{S}} \subseteq [m]$ as*

$$\Delta_{\mathscr{S}}(M) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\mathscr{S}}(M)} \lambda_i \qquad \text{for all } \mathscr{S} \subseteq [m], \tag{2.2}$$

*where*

$$U_{\mathscr{S}}(M) := \Big\{ i \in [n] \colon \sum_{j \in \mathscr{S}^c} m_{ij} = 0 \Big\}$$

*is the subset of service classes that can only be served by servers in $\mathscr{S}$.*

*The menu $M$ admits a steady state under a FCFS-ALIS service discipline if and only if:*

$$\Delta_{\mathscr{S}}(M) > 0 \qquad \text{for all } \mathscr{S} \subseteq [m].$$

**2.2.1 Steady state results for fixed arrival rates** Our results build on the steady state analysis of Adan and Weiss (2014), which we briefly review for completeness. The authors derive their results based on a Markov chain representation of the system defined on a carefully crafted state space $X$. A state in this state space is described by three components: $(i)$ a permutation of servers $s = (s_1, \ldots, s_m)$, $(ii)$ an integer $b \in \{0, \ldots, m\}$ indicating the number of busy servers, and $(iii)$ a vector $(n_1, \ldots, n_b)$ that indicates the composition of customers waiting for service in the different service classes. It is helpful to denote a generic state $x \in X$ by the tuple:

$$x = (s_1, n_1, s_2, n_2, \ldots, s_b, n_b, s_{b+1}, \ldots, s_m). \tag{2.3}$$

The first $b$ components $(s_1, \ldots, s_b)$ of the server permutation $s$ denote the $b$ busy servers

ranked according to the arrival time of the customer they are serving, with server $s_1$ serving the oldest arrival and server $s_b$ serving the youngest arrival. The remaining servers $(s_{b+1}, \ldots, s_m)$ are all idle and ranked in the order they became idle, with $s_{b+1}$ the server that has been idle the longest. Finally, $n_\ell$ for $\ell = 1, \ldots, b$, represents the number of customers in the system who arrived after the job currently being served by $s_\ell$ but before the job currently being served by $s_{\ell+1}$. Due to the FCFS-ALIS service discipline, we know these customers can only be served by some server in $(s_1, \ldots, s_\ell)$. That is, each of these $n_\ell$ customers must belong to some service class in $U(s_1, \ldots, s_\ell)$.

According to (Adan and Weiss, 2014, Theorem 2.1), the steady-state probability of state $x$ admits the product form:

$$\pi(x) = \mathcal{B} \prod_{\ell=1}^{b} \frac{\lambda_{U(s_1,\ldots,s_\ell)}^{n_\ell}}{\mu_{\{s_1,\ldots,s_\ell\}}^{n_\ell+1}} \prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1}, \tag{2.4}$$

where $\mathcal{B}$ is an appropriate normalizing constant. Additionally, each of the $n_\ell$ customers 'between' server $s_\ell$ and server $s_{\ell+1}$ belongs to service class $i \in U(s_1, \ldots, s_\ell)$ independently with probability $\frac{\lambda_i}{\lambda_{U(s_1,\ldots,s_\ell)}}$.

These steady-state probabilities can be used to calculate the expected number of customers of each type in the system. Little's Law can then be applied to calculate expected steady-state mean waiting times. However, if we consider the process for calculating expected waiting times even for our relatively simple example in Figure 2.1, we see that while these calculations are possible, the process is laborious and the resulting expressions are unwieldy. For example, let us consider how we would calculate the expected number of class 4 customers. We first observe that class 4 customers are compatible with all servers. This means that the only times class 4 customers are waiting in the system is if all servers are busy when a class 4 customer arrives. Thus if we want to calculate the expected number of class 4 customers waiting for service in the system, we can restrict ourselves to considering

only the states in which all 4 servers are busy.

Fixing the permutation of servers, and the number of busy servers, the values of $n_i$ are geometrically distributed, and hence the expected values have closed form expressions. For example, if we condition on being in the subset of states $x \in X_{(s_1,s_2,s_3,s_4)}$ such that $b = 4$ and the server permutation $(s_1, s_2, s_3, s_4)$, i.e. $x = (s_1, n_1, s_2, n_2, s_3, n_3, s_4, n_4)$, then the expected value of $n_4$ is

$$\mathbb{E}(n_4 | x \in X_{(s_1,s_2,s_3,s_4)}) = \frac{\mathcal{B}|\lambda| \cdot |\mu|}{(\mu_1 - \lambda_1)(\mu_1 + \mu_2 - (\lambda_1 + \lambda_2))(|\mu| - \mu_4 - (|\lambda| - \lambda_4))(|\mu| - |\lambda|)} \tag{2.5}$$

where $|\lambda| := \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4$, $|\mu| := \mu_1 + \mu_2 + \mu_3 + \mu_4$ and $\mathcal{B}$ is an appropriate normalizing constant. Note that $n_4$ is not the number of class 4 customers; instead $n_4$ is the number of customers who arrived to the system after the customer server 4 is currently serving. Therefore the expected number of class 4 customers conditional on being in the subset of states $X_{(s_1,s_2,s_3,s_4)}$ is $\frac{\lambda_4}{|\lambda|}\mathbb{E}[n_4 | x \in X_{(s_1,s_2,s_3,s_4)}]$.

To fully calculate the expected number of class 4 customers, we would need to repeat this process for every permutation of servers. Since there are four servers, there are 24 possible permutations of servers to sum over, with different combinations of terms appearing in the denominator for each permutation. This gives us very complicated expressions for the expected number of servers. If we were instead looking at the number of class 1 customers, we would also need to consider states in which only some servers are busy, giving us even more server combinations that we need to consider.

It is this underlying computational complexity -which grows combinatorially fast in the size of the system– that motivates our move to heavy traffic. As the system approaches heavy traffic, the probability of being in a state with an idle server approaches 0, letting us restrict our attention only to states in which all servers are busy. Additionally, we show in Proposition 7 that in heavy-traffic, only certain server permutations have positive probability,

which is a fact that simplifies the problem even further.

**2.2.2 Heavy traffic scaling** The last part of the model is the heavy-traffic scaling. As mentioned in the Introduction, our formulation extends Afèche et al. (2021), who consider a specific direction of convergence to heavy traffic to derive their results. Specifically, they assume that the proportions of customers joining the different service classes remain constant as the system approaches heavy traffic. In this work, we allow a general direction of convergence.

We consider a conventional heavy traffic regime in which the arrival rates approach the capacity of the service system, while the number of service classes and servers, and the service menu remain constant. We parameterize our systems by $\epsilon$, and let the service system approach heavy traffic as $\epsilon \downarrow 0$. Specifically, we assume that there is a sequence of arrival rates $\lambda^{(\epsilon)} = \{\lambda_i{}^{(\epsilon)}\}_{i\in[n]}$ where

$$\lambda_i{}^{(\epsilon)} = \Lambda_i - \gamma_i\epsilon + o(\epsilon) \geq 0 \quad \text{for all } i \in [n] \text{ and } 0 < \epsilon < \epsilon_+, \tag{2.6}$$

for some vector $\Lambda \in \mathbb{R}^n_+$, some vector $\gamma \in \mathbb{R}^n$, and some $\epsilon_+ > 0$. We make the following additional assumptions on $\lambda^{(\epsilon)}$ and $\mu$.

**Assumption 1.** *All of the following hold for arrival rates $\lambda^{(\epsilon)}$ given by* (2.6) *and service rates $\mu$:*

(i) $|\Lambda| = |\mu|$,

(ii) $|\gamma| > 0$,

(iii) $\gamma_i < 0$ *for all* $i \in [n]$ *such that* $\Lambda_i = 0$.

Parts (i) and (ii) ensure that we are approaching heavy traffic from below. Part (iii) is implied by $\lambda_i{}^{(\epsilon)} > 0$ for all $i \in [n]$ and $0 < \epsilon < \epsilon_+$, but we include it in Assumption 1 for clarity. Note that for $i \in [n]$ such that $\Lambda_i > 0$, we allow $\gamma_i$ to be positive, negative, or zero.

This is more general than the scaling used in Afèche et al. (2021), where the authors assume that $\gamma = \Lambda$. Additionally, Afèche et al. (2021) requires that $\Lambda_i > 0$ for all $i \in [n]$. We relax that assumption here, as it is useful to allow for no arrivals to particular service classes when considering strategic customer behaviour.

We are only interested in studying systems which produce stable outcomes. This leads us to restrict our attention to a set of *admissible* menus.

**Definition 1.** (Admissible Menus) *For a given menu $M$, arrival rates $\lambda^{(\epsilon)}$, and service rates $\mu$, define for any subset of servers $\mathscr{S} \subseteq [m]$ and $\epsilon > 0$*

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\mathscr{S}}(M)} \lambda^{(\epsilon)}_i. \tag{2.7}$$

*A menu $M$ is* admissible *under arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ if*

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m].$$

In words, this ensures that the menu $M$ and arrival rates $\lambda^{(\epsilon)}$ admit a steady state under a FCFS-ALIS service discipline, and that the slack in the system is converging slowly enough so that the average delays of the different service classes converge when scaled by $\epsilon$.

We let $\mathcal{M}(\lambda^{(\epsilon)}, \mu)$ denote the set of all menus $M$ that are admissible for arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$. The set $\mathcal{M}(\lambda^{(\epsilon)}, \mu)$ will be non-empty for all pairs $(\lambda^{(\epsilon)}, \mu)$ satisfying Assumption 1. To see this, observe that the complete menu $M$ such that $m_{ij} = 1$ for all $i \in [n]$ and $j \in [m]$ will be admissible for all $(\lambda^{(\epsilon)}, \mu)$ satisfying Assumption 1. The complete menu will operate like a single queue with arrival rates $|\lambda^{(\epsilon)}|$ that is served by all servers.

## 2.3 Mean Waiting Times in Heavy Traffic

We are interested in calculating the mean waiting times of the different service classes. Because we are looking at a conventional heavy traffic setting, the waiting times themselves will grow out of bound as $\epsilon \downarrow 0$. We will instead look at the scaled mean waiting times

$$\widehat{W_i}^{(\epsilon)} = \epsilon \cdot W_i{}^{(\epsilon)}, \tag{2.8}$$

which will remain bounded in heavy traffic.

In what follows we show how to find the limiting expected waiting times by building upon and extending the methods and results in Afèche et al. (2021).

**2.3.1 Feasible flows and CRP components** We begin by identifying the feasible flows of customers between service classes and servers. For arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ satisfying Assumption 1, and an admissible menu $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$, for $0 \leq \epsilon < \epsilon_0$ we define the set of feasible flows as

$$\begin{aligned}\mathcal{F}(\epsilon, \lambda^{(\epsilon)}, M) := \Big\{ f = [f_{ij}] \geq 0 : & \sum_{i \in [n]} f_{ij} \leq \mu_j, \quad \forall j \in [m]; \\ & \sum_{j \in [m]} f_{ij} = \lambda_i{}^{(\epsilon)}, \quad \forall i \in [n]; \quad f_{ij} = 0, \quad \forall (i,j) : m_{ij} = 0 \Big\},\end{aligned} \tag{2.9}$$

where $\epsilon_0 \in \mathbb{R}$ is such that $\lambda^{(\epsilon)} > 0$ for all $0 < \epsilon < 0$. We know from the admissibility of $M$ that such an $\epsilon_0$ exists, and that $\mathcal{F}(\epsilon, \lambda^{(\epsilon)}, M)$ is non-empty for all $0 < \epsilon < \epsilon_0$. The following lemma shows that $\mathcal{F}(0, \lambda^{(\epsilon)} M)$ is also non-empty. The proof relies on $\mathcal{F}(\epsilon, \lambda^{(\epsilon)}, M)$ being a subset of a compact set $\mathcal{F}_{\max}(\lambda^{(\epsilon)})$ for $0 \leq \epsilon < \epsilon_0$.

**Lemma 1.** *For a given $\lambda^{(\epsilon)}$ and $\mu$ satisfying Assumption 1, and $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$, the*

*set* $\mathcal{F}(0,\Lambda,M)$ *is non-empty. Furthermore, every sequence of flows* $f^{(\epsilon)}$ *such that* $f^{(\epsilon)} \in \mathcal{F}(\epsilon,\lambda^{(\epsilon)},M)$ *has a sub-sequence that converges to some* $\tilde{f} \in \mathcal{F}(0,\Lambda,M)$.

PROOF: See Appendix 2.A. □

As this lemma suggests, the set $\mathcal{F}(0,\Lambda,M)$ contains information about what sort of flows it is possible to observe in heavy traffic. We will use the set of feasible limiting flows to determine which servers have a positive probability of serving which service classes in the limit. To do this, we will first define the *residual matching* of the menu $M$.

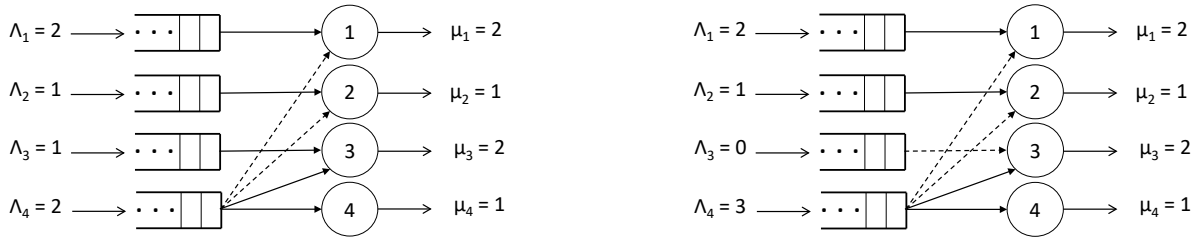
**Definition 2.** (Residual Matching) *For a given* $(\lambda^{(\epsilon)},\mu,M)$ *such that* $\lambda^{(\epsilon)}$ *and* $\mu$ *satisfy Assumption 1 and* $M \in \mathcal{M}(\lambda^{(\epsilon)},\mu)$ *we define the* residual matching $\breve{M}$*, where* $\breve{M} = [\breve{m}_{ij}]$ *satisfies* $\breve{m}_{ij} = 1$ *if and only if there exists flows* $\tilde{f} \in \mathcal{F}(0,\Lambda,M)$ *such that* $\tilde{f}_{ij} > 0$.

Intuitively, for a service class $i$ and server $j$ with $m_{ij} = 1$ but $\breve{m}_{ij} = 0$, the flow of customers from service class $i$ to server $j$ must vanish in the heavy-traffic limit. Afèche et al. (2021) provide an algorithm for finding the residual matching. However, for small, simple systems the residual matching can be found by inspection. To see this, consider again the simple example in Figure 2.1, specifying the service rates to be $\mu = [2,1,2,1]$. We will consider two example vectors of arrival rates, $\Lambda_a = [2,1,1,2]$ and $\Lambda_b = [2,1,0,3]$. In each case, there is only one set of feasible flows in $\mathcal{F}(0,\Lambda_a,M)$ and $\mathcal{F}(0,\Lambda_b,M)$, given by

$$f^a_{ij} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad f^b_{ij} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix}. \tag{2.10}$$

In example (a), the arcs in the compatibility network with $m_{ij} = 1$ and $\breve{m}_{ij} = 0$ are (4,1) and (4,2). While service class 4 is compatible with servers 1 and 2, there will be zero

flow between class 4 and servers 1 and 2 in the limit. All the service capacity of servers 1 and 2 will be allocated to serving classes 1 and 2. We can see this visually in panel (a) in Figure 2.2, where the arcs with $m_{ij} = 1$ and $\breve{m}_{ij} = 1$ are represented with solid lines, and the arcs with $m_{ij} = 1$ and $\breve{m}_{ij} = 0$ are represented with dashed lines. Example (b) is similar, but we now additionally have arc (3,3) with $m_{33} = 1$ and $\breve{m}_{33} = 0$. In panel (b) of Figure 2.2 we can see that class 3 only has one dashed arc connecting it to any servers, representing that no servers are allocating any capacity to class 3 in the limit, even though class 3 is compatible with server 3.




(a) Residual matching (a)

(b) Residual matching (b)

Figure 2.2: Examples of residual matchings.

Knowing the residual matching allows us to decompose the initial bipartite matching system into a partition of independent components, which Afèche et al. (2021) refer to as *complete resource pooling* (CRP) components.

**Definition 3.** (CRP Component) *For a given* $(\lambda^{(\epsilon)}, \mu, M)$ *such that* $\lambda^{(\epsilon)}$ *and* $\mu$ *satisfy Assumption 1 and* $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$*, let the induced residual matching be denoted* $\breve{M}$*. We say that the subset* $\mathbb{C} = (\mathcal{C}, \mathcal{S}) \in 2^{[n]} \times 2^{[m]}$ *of service classes and servers forms a* CRP component *if for any pair of nodes* $k_1, k_2 \in \mathcal{C} \cup \mathcal{S}$ *there exists a path between* $k_1$ *and* $k_2$ *in* $\breve{M}$*, and* $\mathbb{C}$ *is maximal in the sense that the condition is violated for any strict superset of* $\mathbb{C}$*.*

We let $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ denote the collection of CRP components induced by the residual matching $\breve{M}$, where $K$ is the number of components. Each $\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k)$ is defined by

the subset of service classes $\mathcal{C}_k$ and the subset of servers $\mathcal{S}_k$ that belong to $\mathbb{C}_k$. Since we allow for service classes with no arrivals, that is $\Lambda_i = 0$, some CRP components will have an empty server set. Each service class with $\Lambda_i = 0$ forms a separate CRP component with an empty server set. We denote the subset of such CRP components by $\mathcal{I}_0$:

$$\mathcal{I}_0 = \{k : \Lambda_k = 0\}. \tag{2.11}$$

We let $K' := K - |\mathcal{I}_0|$ be the number of CRP components with non-empty sets of servers, and will assume that the CRP components are indexed so that the components in $[K] \setminus \mathcal{I}_0$ have indices $1, 2, \ldots, K'$. We will use $k(i)$ and $k(j)$ to denote the component that service class $i$ or server $j$ is part of, where the use should be clear from context.

To make these ideas more concrete, let us return to our examples in Figure 2.2. In example (a), service class 1 and server 1 make up a CRP component, as they are not connected to any other service classes or servers with solid arcs. Similarly, service class 2 and server 2 make up a CRP component. We can see a path between classes 3 and 4 through server 3, so these classes along with servers 3 and 4 make up a single CRP component. This means the CRP components for example (a) can be written as $\mathbb{C}_1 = (\mathcal{C}_1, \mathcal{S}_2 = (\{1\}, \{1\})$, $\mathbb{C}_2 = (\mathcal{C}_2, \mathcal{S}_2 = (\{2\}, \{2\})$, and $\mathbb{C}_3 = (\mathcal{C}_3, \mathcal{S}_3 = (\{3, 4\}, \{3, 4\})$. Example (b) is similar, the difference being that now service class 3 is not connected to any server or service class with a solid arc, and therefore is in a CRP component by itself with an empty server set, i.e. $\mathcal{I}_0 = \{3\}$. So the CRP components for example (b) are $\mathbb{C}_1 = (\mathcal{C}_1, \mathcal{S}_2 = (\{1\}, \{1\})$, $\mathbb{C}_2 = (\mathcal{C}_2, \mathcal{S}_2 = (\{2\}, \{2\})$, $\mathbb{C}_3 = (\mathcal{C}_3, \mathcal{S}_3 = (\{4\}, \{3, 4\})$, and $\mathbb{C}_4 = (\mathcal{C}_4, \mathcal{S}_4 = (\{3\}, \{\emptyset\})$.

Abusing notation, we denote the aggregate arrival and service rates for the CRP components under $\lambda^{(\epsilon)}$ as:

$$\forall k \in [K] \ : \ \widetilde{\lambda}_k^{(\epsilon)} = \sum_{i \in \mathcal{C}_k} {\lambda_i}^{(\epsilon)} =: \widetilde{\Lambda}_k - \epsilon \widetilde{\gamma}_k + o(\epsilon), \quad \text{and} \quad \widetilde{\mu}_k = \sum_{j \in \mathcal{S}_k} \mu_j, \tag{2.12}$$

where $\widetilde{\Lambda}_k = \sum_{i \in \mathcal{C}_k} \Lambda_i$ and $\widetilde{\gamma}_k = \sum_{i \in \mathcal{C}_k} \gamma_i$. We will later show that each CRP component must satisfy $\widetilde{\Lambda}_k = \widetilde{\mu}_k$ so that the slack between demand and capacity within a CRP component in heavy-traffic goes to zero with $\epsilon$. While each CRP component is critically loaded, the "well-connectedness" within a CRP component allows shifting load from one service class to another on short time scales. In particular, we will show in Theorem 1 that under an FCFS-ALIS policy, waiting times are balanced in such a way that service classes that belong to the same CRP component have the same limiting scaled mean waiting time in the heavy traffic limit.

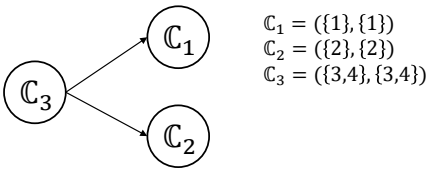
**2.3.2 Directed Acyclic Graph of CRP components** The menu $M$ and the residual matching $\breve{M}$ uniquely induce a directed acyclic graph (DAG) on the collection of CRP components defined in the previous step. This is useful as the DAG defines a precedence relation among service classes: since component $k_1$ has a directed arc to component $k_2$, there is a service class in $k_1$ that can be served by a server in $k_2$. This means $k_1$ can "off-load" its customers to the servers of component $k_2$, and so the instantaneous waiting time in component $k_1$ cannot exceed that in component $k_2$ under FCFS-ALIS. This intuition is made precise in the proof of Theorem 1.

The following is a formal statement of how the DAG is induced.

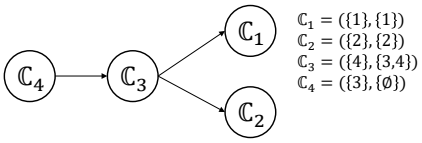
**Definition 4.** (DAG) *Given the menu $M = [m_{ij}]$, and the CRP components $\{\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k) \colon k = 1, \ldots, K\}$ induced by the residual matching $\breve{M}$, we define $\mathcal{D} = ([K], \mathcal{A})$ associated to $M$ as the directed acyclic graph whose nodes correspond to the CRP components, and there is a directed arc $(k_1, k_2) \in \mathcal{A}$ from component $\mathbb{C}_{k_1}$ to component $\mathbb{C}_{k_2}$ if and only if there exists a service class $i \in \mathcal{C}_{k_1}$ and a server $j \in \mathcal{S}_{k_2}$ such that $m_{ij} = 1$. We use the notation $k_1 \overset{\mathcal{D}}{\rightsquigarrow} k_2$ to denote that there is a directed path $k_1$ to $k_2$ in the DAG $\mathcal{D}$.*

(Afèche et al., 2021, Lemma 2) formally proves that the directed graph defined above is in fact acyclic.

Returning to our examples in Figure 2.2, the DAGs are given below. In both cases,




(a) DAG (a)




(b) DAG (b)

Figure 2.3: Examples of DAGs.

service class 4 can be served by servers 1 and 2 in the original menu, i.e. $m_{41} = m_{42} = 1$, and so there are directed arcs from $\mathbb{C}_3$ to $\mathbb{C}_1$ and $\mathbb{C}_2$. In example (b), $\mathbb{C}_4$ contains service class 3 but no servers, since service class 3 has an arrival rate of 0. Therefore $\mathbb{C}_4$ has a directed arc to $\mathbb{C}_3$, as this is the CRP component containing the server that service class 3 is compatible with.

As we mentioned earlier, our computations for the heavy-traffic waiting times build on the work of Adan and Weiss (2014). The crucial component of their analysis is a state-space representation for the FCFS-ALIS matching model which involves ranking the busy servers in order of the waiting time of the customers they are serving. As was proved in Afèche et al. (2021) for the less general scaling, in heavy-traffic this entails restricting attention to only certain permutations of the CRP components which have asymptotically non-zero steady-state probability. We show in Proposition 7 that this also holds for our more general scaling. The topological orders of the DAG $\mathcal{D}$ are precisely these permutations. The definition we give next differs slightly from Afèche et al. (2021) due to the potential presence of CRP components with $\widetilde{\Lambda}_k = 0$.

**Definition 5.** (Topological Orders on CRP Components) *Let* $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_{K'}\}$ *be the CRP components with* $\widetilde{\Lambda}_k > 0$. *Given the DAG* $\mathcal{D} = ([K], \mathcal{A})$, *we say that a permutation* $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(K'))$ *of* $[K']$ *induces a* topological order $(\mathbb{C}_{\sigma(1)}, \mathbb{C}_{\sigma(2)}, \ldots, \mathbb{C}_{\sigma(K')})$ *of these CRP components if for every pair* $(k_1, k_2) \in [K']$ *such that* $k_1 \overset{\mathcal{D}}{\rightsquigarrow} k_2$, *we have* $\sigma^{-1}(k_2) <$

$\sigma^{-1}(k_1)$. *In other words, sink components of* $\mathcal{D}$ *precede source components. We let* $\mathcal{T}(\mathcal{D}, K')$ *denote the set of all permutations* $\sigma$ *of* $[K']$ *that induce a topological order on components* $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$.

*Further, for each* $\sigma \in \mathcal{T}(\mathcal{D}, K')$*, we partition the CRP components* $[K]$ *by associating a subset for each* $k \in [K']$ *as follows:*

$$\mathsf{comps}(\sigma, k) := \{\sigma(k)\} \cup \{\kappa \in \mathcal{I}_0 : k = \max\{k' \in [K'] : \ \kappa \overset{\mathcal{D}}{\rightsquigarrow} \sigma(k')\}\}. \tag{2.13}$$

*The interpretation of this is that for each index* $k \in [K']$*, we associate the CRP component corresponding to* $\sigma(k)$ *as well as all CRP components* $\kappa$ *with* $\widetilde{\Lambda}_\kappa = 0$ *(i.e., server-less components) for which the component* $\sigma(k)$ *is the last component in the topological order* $\sigma$ *that is reachable from* $\kappa$ *via a directed path.*

We will use the shorthand $\mathsf{comps}^{-1}(\sigma, k)$ to denote the index $\kappa \in [K']$ such that $k \in \mathsf{comps}(\sigma, \kappa)$.

To highlight the difference with Afèche et al. (2021), under the heavy-traffic regime considered in Afèche et al. (2021) all CRP components have a non-empty server set $\mathcal{S}_j$. In contrast, in our model, we have service classes that are in CRP components by themselves. These CRP components are special in that they have no incoming arc in the DAG $\mathcal{D}$, and can only have a directed arc to CRP components with non-empty server sets. The topological orders $\mathcal{T}(\mathcal{D}, K')$ can thus be thought of as preprocessing $\mathcal{D}$ to remove the server-less CRP components $\{\mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ which are "hanging off" $\mathcal{D}$, and finding topological orders on the remaining components. Since the topological order has sink components of $\mathcal{D}$ preceding source components, and as we mentioned earlier, the DAG defines a precedence relation among service classes, we can then interpret $\mathsf{comps}^{-1}(\sigma, k)$ as associating each server-less CRP component with the CRP component that is reachable from it that has the shortest steady-state wait.

Returning to our examples in Figure 2.3, both example (a) and example (b) have the same set of CRP components with positive limiting arrival rates, the set $\{\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3\}$. Both examples also have the same connectivity with these components. $\mathbb{C}_3$ has directed arcs to $\mathbb{C}_1$ and $\mathbb{C}_2$, but there are no arcs between $\mathbb{C}_2$ and $\mathbb{C}_2$. Hence in any topological orders on these CRP components, we know that $\mathbb{C}_1$ and $\mathbb{C}_2$ come before $\mathbb{C}_3$, but $\mathbb{C}_1$ can come either before or after $\mathbb{C}_2$. Thus the possible permutations are $\sigma_1 = (1, 2, 3)$ and $\sigma_2 = (2, 1, 3)$, and the associated topological orders are $(\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3)$ and $(\mathbb{C}_2, \mathbb{C}_1, \mathbb{C}_3)$. As example (a) has no CRP components with limiting arrival rates of 0, for each $\sigma$ and each $k$, $\mathsf{comps}(\sigma, k)$ is simply the set containing the index of the CRP component at position $k$ of the topological order $\sigma$. In example (b), $\mathbb{C}_4$ has $\tilde{\lambda}_4 = 0$, so for each topological order $\sigma$, we need to determine for which $k$ we have $4 \in \mathsf{comps}(\sigma, k)$. The only directed arc from $\mathbb{C}_4$ to any other CRP component is to $\mathbb{C}_3$. Hence for each $\sigma$, we have that $4 \in \mathsf{comps}(\sigma, k)$ if and only if $3 \in \mathsf{comps}(\sigma, 4)$. Since $\mathbb{C}_3$ is the last element of the topological order for both permutations $\sigma_a$ and $\sigma_b$, we have that $\mathsf{comps}(\sigma_a, 3) = \mathsf{comps}(\sigma_b, 3) = \{3, 4\}$.

**2.3.3 Calculating waiting times** Let $\mathcal{T}(\mathcal{D}, K') = (\sigma_1, \ldots, \sigma_T)$ be the collection of topological orders on $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$ (the components with $\widetilde{\Lambda}_k > 0$). For a topological order $\sigma_t \in \mathcal{T}(\mathcal{D}, K')$ with the associated function $\mathsf{comps}(\sigma_t, \cdot)$ defined in (2.13), we define the unnormalized probability of being in a state associated with the topological order $\sigma_t$ as:

$$\mathbb{Q}(\sigma_t) = \prod_{\kappa \in [K']} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t, \ell)}}, \tag{2.14}$$

where we use the shorthand

$$\widetilde{\gamma}_{\mathsf{comps}(\sigma, \ell)} = \sum_{\kappa \in \mathsf{comps}(\sigma, \ell)} \widetilde{\gamma}_{\kappa}.$$

For a permutation $\sigma_t \in \mathcal{T}(\mathcal{D}, K')$, for any CRP component $\mathbb{C}_k$, we define the waiting time conditioned on the topological order $\sigma_t$ as:

$$w_{\sigma_t,k} = \sum_{\kappa=\mathsf{comps}^{-1}(\sigma_t,k)}^{K'} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)}}. \tag{2.15}$$

The following Lemma 2 proves that the expressions above are well-defined.

**Lemma 2.** *For $\lambda^{(\epsilon)}$ and $\mu$ satisfying Assumption 1, and for some $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$ for all permutations $\sigma_t \in \mathcal{T}(\mathcal{D}, K')$ of CRP components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$ and for all $\kappa \in [K']$,*

$$\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)} > 0.$$

Proof: See Appendix 2.A. □

With the expressions for the unnormalized probabilities and conditional waiting times of topological orders in place, we are ready to state our main theorem regarding the mean scaled steady-state waiting times of different service classes.

**Theorem 1.** *Take any $(\lambda^{(\epsilon)}, \mu, M)$ such that $\lambda^{(\epsilon)}$ and $\mu$ satisfy Assumption 1. For an admissible menu $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$, let $\breve{M}$ be the residual matching, and let the collection of CRP components induced by $\breve{M}$ be denoted $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$. Then, service classes that belong to the same CRP component experience the same scaled steady-state mean waiting time in heavy traffic. Furthermore, the scaled steady-state mean waiting time of CRP component $\mathbb{C}_k$ is equal to*

$$\widehat{W}_{\mathbb{C}_k} = \sum_{t=1}^{T(M)} \left( \frac{\mathbb{Q}(\sigma_t)}{\mathbb{Q}(\sigma_1) + \mathbb{Q}(\sigma_2) + \cdots + \mathbb{Q}(\sigma_{T(M)})} \right) w_{\sigma_t,k}. \tag{2.16}$$

The proof of Theorem 1 can be found in Section 2.6.1.

## 2.4 Matching Probabilities in Heavy Traffic

Another performance metric of interest is the matching probabilities, that is, for each service class $i$ and server $j$, the probability that a customer who joins class $i$ is served by server $j$. For any menu $M$ that is admissible with arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$, we let $p^{(\epsilon)}(M, \lambda^{(\epsilon)}, \mu)$ be the matrix of matching probabilities, so $p_{ij}^{(\epsilon)}(M, \lambda^{(\epsilon)}, \mu)$ is the steady state probability with which a customer who joins class $i \in [n]$ is served by server $j \in [m]$. While exact matching probabilities are difficult to calculate, and remain difficult to calculate even in heavy traffic, we are able to provide two results regarding how matching rate calculations simplify as we move to heavy traffic.

Before stating our results, it will be useful to describe the combinations of limiting arrival rates $\Lambda$, service rates $\mu$, and menus $M$ such that there is some sequence $\lambda^{(\epsilon)}$ converging to $\Lambda$ that makes $M$ admissible. The following proposition will help us understand these combinations.

**Proposition 2.** *Take any sequence of arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ such that , $\lambda^{(\epsilon)}$ and $\mu$ satisfy Assumption 1, and let $M$ be such that $M \in \mathcal{M}(\lambda^{(\epsilon)}\mu)$. Let $\Lambda = \lim_{\epsilon \to 0} \lambda^{(\epsilon)}$. Then $M$ is admissible with service rates $\mu$ and arrival rates*

$$\lambda^{(\epsilon)} = \Lambda - \epsilon\Lambda, \quad \textit{for } \epsilon > 0.$$

*Furthermore, if $M$ is admissible with $\lambda^{(\epsilon)} = \Lambda - \epsilon\Lambda$ and $\mu$, then the menu $\breve{M}$ given by the residual matching of $M$ is also admissible with $\lambda^{(\epsilon)} = \Lambda - \epsilon\Lambda$ and $\mu$.*

Proof: See Appendix 2.B. □

This lets us talk about menus that are admissible for limiting arrival rates $\Lambda$ and service rates $\mu$. We will define the set $\mathcal{M}^+(\Lambda, \mu)$ to be the set of all menus $M$ such that $M$ is admissible for arrival rates $\lambda^{(\epsilon)} = \Lambda(1 - \epsilon)$ and service rates $\mu$. This provides us with a more

convenient way to express our results regarding matching probabilities, the first of which is stated formally in Theorem 2. This tells us that while the limiting expected delays depend on the particular sequence of arrival rates $\lambda^{(\epsilon)}$, and in particular depend on the slacks $\gamma$, the matching probabilities depend only on the limiting arrival rates.

**Theorem 2.** *Take any limiting arrival rates $\Lambda$ and service rates $\mu$ such that $|\Lambda| = |\mu|$. Consider any menu $M \in \mathcal{M}^+(\Lambda, \mu)$. Take any two sequences of arrival rates $\lambda_a^{(\epsilon)}$ and $\lambda_b^{(\epsilon)}$ such that $\lim_{\epsilon\to 0} \lambda_a^{(\epsilon)} = \lim_{\epsilon\to 0} \lambda_b^{(\epsilon)} = \Lambda$, both sequences satisfy Assumption 1 with $\mu$, and $M$ is admissible for both sequences of arrival rates with $\mu$. Then $\lim_{\epsilon\to 0} p_{ij}^{(\epsilon)}(M, \lambda_a^{(\epsilon)}, \mu) = \lim_{\epsilon\to 0} p_{ij}^{(\epsilon)}(M, \lambda_b^{(\epsilon)}, \mu)$ for all $i \in [n]$ and $j \in [m]$.*

The proofs of Theorem 2 and Corollary 1 can be found in Section 2.6.2.

Theorem 2 lets us talk about the matching probabilities of a menu $M$ just in terms of the limiting arrival rates $\Lambda$ and service rates $\mu$. In light of this, for the rest of this chapter, we will refer to matching probabilities in terms of the limiting arrival rates, that is, we will write $p_{ij}^{(\epsilon)}(M, \Lambda, \mu)$.

The second result we have relating to matching probabilities, stated formally in Corollary 1, tells us that matching probabilities within a CRP component are independent of all other CRP components.

**Corollary 1.** *Take any limiting arrival rates $\Lambda$ and service rates $\mu$ such that $|\Lambda| = |\mu|$, and take any $M \in \mathcal{M}^+(\Lambda, \mu)$. Let $\breve{M}$ be the residual matching, and let the collection of CRP components induced by $\breve{M}$ be denoted $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$. Then for any service class $i \in \mathcal{C}_k$ and server $j \in \mathcal{S}_k$,*

$$\lim_{\epsilon\to 0} p_{ij}^{(\epsilon)}(M, \Lambda, \mu) = \lim_{\epsilon\to 0} p_{ij}^{(\epsilon)}(\breve{M}, \Lambda, \mu).$$

Corollary 1 implies that when calculating the matching rates, we can look at each CRP component individually. Additionally, it tells us that the DAG structure does not affect the

matching probabilities. We will see in Section 2.5 that two menus $M$ and $M'$ with the same residual matching $\breve{M}$ can have significantly different expected waiting times in heavy-traffic if the two menus induce different DAGs. Corollary 1 tells us that despite this, the limiting matching probabilities of menus $M$ and $M'$ are the same.

## 2.5 Discussion

Before getting into the proofs of our main results, we discuss some of their implications, while highlighting the differences between the behaviours of our model and the model in Afèche et al. (2021). We also explore some simple questions regarding the design of menus of service classes.

**2.5.1 Implementable outcomes** Our motivation for the heavy-traffic scaling used in this dissertation is that it allows for a wider range of outcomes than the proportional scaling used in Afèche et al. (2021). The following definition will help formalise what we mean by this.

**Definition 6.** (Implementable Waiting Times) *Take limiting arrival rates $\Lambda$, service rates $\mu$, and a menu $M$ such that a collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ is induced. We say a vector of limiting scaled waiting times $W = (W_1, W_2, \ldots, W_K)$ is* implementable *if there exists $\gamma \in \mathbb{R}^n$ such that the menu $M$ is admissible for the pair $(\lambda^{(\epsilon)}, \mu)$ where*

$$\lambda_i^{(\epsilon)} = \Lambda_i - \epsilon\gamma_i + o(\epsilon), \quad \text{for all } i \in [n],$$

*and the resulting limiting waiting times $\widehat{W}_{\mathbb{C}_k}$ given by (2.16) are equal to $W_k$ for all $k \in [K]$.*

If we only look at the scaling in Afèche et al. (2021), in which $\gamma = \Lambda$, then each combination of limiting arrival rates $\Lambda$, service rates $\mu$, and menu $M$ can produce one specific vector of waiting times. By allowing $\gamma$ to change, we increase the set of implementable outcomes.

As we alluded to in Section 2.3, the DAG provides information about which vectors of waiting times are implementable. The following statement, which is a corollary of Theorem 1, formalises this idea.

**Corollary 2.** *If $W \in \mathbb{R}_+^K$ is implementable, then $W$ is consistent with some topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$. That is, there is some topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$ such that $W_k \leq W_\kappa$ only if $\mathsf{comps}^{-1}(\sigma, \kappa) \leq \mathsf{comps}^{-1}(\sigma, k)$.*

PROOF: See Appendix 2.C. □

Corollary 2 provides a necessary condition for waiting times to be implementable. While completely characterising the set of implementable waiting times for a particular $\Lambda$, $\mu$, and $M$ is difficult in general, we are able to provide a sufficient condition for waiting times to be implementable for menus such that the DAG satisfies the following property.

**Definition 7.** (Chained DAGs) *A DAG on $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ is* chained *if there exists a partition $\mathscr{C} = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_L\}$ of $\mathbb{C}$ such that the DAG includes a directed arc from $\mathbb{C}_i$ to $\mathbb{C}_k$ if and only if $\mathbb{C}_i \in \mathscr{C}_\ell$ and $\mathbb{C}_k \in \mathscr{C}_{\ell+1}$ for some $\ell \in [L-1]$.*

Figure 2.4 illustrates an example of a chained DAG in panel (a) and one unchained DAG (i.e., a DAG that is not chained) in panel (b), both over a collection of seven CRP components. For the chained DAG in panel (a), $L = 4$ and $\mathscr{C}_1 = \{\mathbb{C}_2, \mathbb{C}_3\}$, $\mathscr{C}_2 = \{\mathbb{C}_4\}$,

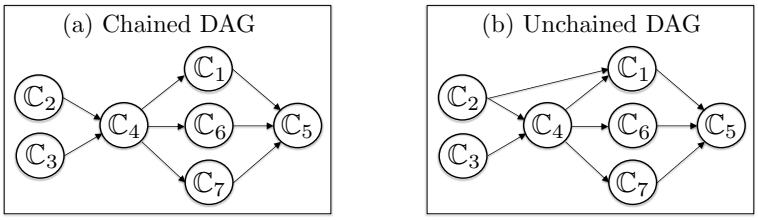



Figure 2.4: Examples of chained (panel a) and unchained (panel b) DAGs over seven CRP components.

$\mathscr{C}_3 = \{\mathbb{C}_1, \mathbb{C}_6, \mathbb{C}_7\}$ and $\mathscr{C}_4 = \{\mathbb{C}_5\}$. On the other hand, to see that the DAG in panel (b) is

not chained, note that we cannot satisfy the requirement in Definition 7 if we consider the three CRP components $\mathbb{C}_1$, $\mathbb{C}_2$ and $\mathbb{C}_4$. Indeed, the arcs connecting $\mathbb{C}_2$ and $\mathbb{C}_4$ to $\mathbb{C}_1$ would require that $\mathbb{C}_2$ and $\mathbb{C}_4$ belong to the same class $\mathscr{C}_l$ in the partition $\mathscr{C}$ for some $\ell$, but then the arc connecting $\mathbb{C}_2$ to $\mathbb{C}_4$ would require these two CRP components to be in different classes in $\mathscr{C}$.

For menus such that the DAG is chained, the following result regarding which vectors of waiting times are implementable applies.

**Proposition 3.** *Take limiting arrival rates $\Lambda$, service rates $\mu$, and a menu $M$ such that $M \in \mathcal{M}_+(\Lambda, \mu)$, and the collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ and the chained DAG $\mathcal{D} = (|K|, \mathcal{A})$ are induced. Let $\mathscr{C} = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_L\}$ be the partition of $\mathbb{C}$ described in Definition 7.*

*The vector $W = (W_1, W_2, \ldots, W_K) \in \mathbb{R}_+^K$ is implementable if the following both hold:*

(i) *$W_k = W_\kappa$ for all $(k, \kappa) \in [K] \times [K]$ such that $W_k \in \mathscr{C}_\ell$ and $W_\kappa \in \mathscr{C}_\ell$ for some $\ell \in [L]$,*

(ii) *$W_k < W_\kappa$ for all $(k, \kappa) \in [K] \times [K]$ such that $W_k \in \mathscr{C}_\ell$ and $W_\kappa \in \mathscr{C}_{\ell'}$ for some $(\ell, \ell') \in [L] \times [L]$ where $\ell < \ell'$.*

Proof: See Appendix 2.C. □

This tells us that we greatly increase the set of implementable outcomes by using a more general heavy traffic scaling.

**2.5.2 Menu Design** We now turn our attention to some simple questions regarding the design of menus of service classes. We will consider two objectives: (1) minimising the total average delay across all service classes, and (2) minimising the maximum expected delay of any service class. We will assume that the arrival rates into the service classes $\lambda^{(\epsilon)}$ and the service rates $\mu$ are fixed, and the service provider is designing the menu $M$, or the compatibility between the service classes and servers.

When the service provider has complete flexibility over how to design the menu, the service provider can minimise both the average delay and the maximum delay faced by any service class simultaneously. The following proposition shows that this can be achieved with a menu that has a single CRP component.

**Proposition 4.** *Given arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ satisfying Assumption 1, for any admissible menu $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$,*

$$\widehat{W}_{\mathbb{C}_k} \geq \frac{1}{|\Gamma|},$$

*for all $k \in [K]$.*

*Furthermore, $\widehat{W}_{\mathbb{C}_k} = \frac{1}{|\Gamma|}$ for some $k \in [K]$ if and only if there exists a directed path from $\widehat{W}_{\mathbb{C}_k}$ to any other CRP component $\mathbb{C}_\kappa$ with $\kappa \in \{[K] \setminus k\}$. This condition is trivially satisfied if there is only one CRP component.*

Proof: See Appendix 2.C. □

Therefore any menu that induces a single CRP component will ensure that all service classes achieve the minimum possible expected delay, hence minimising both the average delay across all service classes and the maximum delay faced by any service class. The following proposition is helpful in designing such a menu.

**Proposition 5.** *Consider a system with limiting arrival rates $\Lambda$ and service rates $\mu$. Any menu $M$ such that*

$$\sum_{j \in \mathscr{S}} \sum_{i \in [n]} \Lambda_i m_{ij} < \sum_{j \in \mathscr{S}} \mu_j, \quad \text{for all } \mathscr{S} \subsetneq [m]$$

*will be admissible for any vector of slacks $\Gamma \in \mathbb{R}^n$ such that $|\Gamma| > 0$. Furthermore, such a menu will induce a single CRP component.*

Proof: See Appendix 2.C. □

A complete menu, in which every service class is compatible with every server, will always satisfy this condition. The complete menu will operate like a single queue served by all servers

according to an FCFS service discipline. Proposition 5 also tells us that we do not need to know the values for the slacks $\Gamma$ to design a delay minimising menu, making it easier to implement in practice.

While a menu that induces a single CRP component minimises delays, it may not be desirable or even feasible to offer such a menu due to real-world compatibility constraints on which servers can serve which customer types. Motivated by these sorts of constraints, we consider the question of how to design the DAG on a collection of CRP components to minimise expected delays for customers.

It will be useful first to understand the expression for average expected delays across all service classes. In Equation (2.16) we defined the delay of each CRP component conditional on being in a particular topological order. We can similarly define $\bar{w}_\sigma$, the average delay across all service classes conditional on being in a particular topological order $\sigma$, as

$$\bar{w}_\sigma = \sum_{\kappa=1}^{K'} \frac{\sum_{k=1}^{\kappa} \widetilde{\mu}_{\sigma(k)}}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}. \tag{2.17}$$

This then lets us express the average expected delay for a particular menu $M$ as

$$\bar{W} = \frac{1}{|\mu|} \sum_{t=1}^{T(M)} \left( \frac{\mathbb{Q}(\sigma_t)}{\mathbb{Q}(\sigma_1) + \mathbb{Q}(\sigma_2) + \cdots + \mathbb{Q}(\sigma_{T(M)})} \right) \sum_{\kappa=1}^{K'} \frac{\sum_{k=1}^{\kappa} \widetilde{\mu}_{\sigma(k)}}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)}}. \tag{2.18}$$

Here we can also see the differences with Afèche et al. (2021), in which the authors find that the average delays depend only on the number of CRP components. With our more general scaling, the average delays depend on the values of the slacks themselves, as well as the structure of the DAG and the set of topological orders that are induced.

Introducing additional arcs into the DAG reduces the number of topological orders. If we can introduce or remove arcs from a DAG in such a way that the system spends more time in states associated with topological orders that have lower conditional average delays

$\bar{w}_\sigma$, then the total average delay will be reduced. However, the values of the slacks of the different CRP components $\widetilde{\gamma}$ limit how we are able to adjust the DAG and still have an admissible menu. This leads us to the following definition of an admissible topological order.

**Definition 8.** *A topological order $\sigma$ is* admissible *for arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ satisfying Assumption 1, and a collection of CRP components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ if and only if $\sum_{\ell=1}^{k} \widetilde{\gamma}_\ell > 0$ for all $k \in [K']$.*

The following lemma tells us how admissible topological orders relate to admissible menus.

**Lemma 3.** *Take any arrival rates $\lambda^{(\epsilon)}$ and service rates $\mu$ satisfying Assumption 1, and any collection of CRP components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$. For any admissible topological order $\sigma$, we can construct an admissible menu $M \in \mathcal{M}(\lambda^{(\epsilon)}, \mu)$ such that the DAG induced by $M$ with $\lambda^{(\epsilon)}$ and $\mu$ only admits the topological order $\sigma$. Furthermore, if $\sigma$ is not admissible, then there are no admissible menus $M$ that admit the topological order $\sigma$.*

Proof: See Appendix 2.C. □

The set of admissible topological orders tells us which DAGs are feasible given a particular CRP component. We can then minimise average delays by identifying the topological order with the lowest condition delays.

**Proposition 6.** *Given limiting arrival rates $\Lambda$, service rates $\mu$, slacks $\Gamma$, and CRP components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$, there will be a permutation of CRP components $\sigma$ that minimises the average expected delay across all implementable topological orders,*

$$\bar{w}_\sigma = \sum_{\kappa=1}^{K'} \frac{\sum_{k=1}^{\kappa} \widetilde{\mu}_{\sigma(k)}}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}.$$

*The DAG or menu that will minimise delays is one that only allows for this topological order.*

Proof: See Appendix 2.C. □

Given that adding arcs to a DAG is achieved by adding additional flexibility to a service system, one might think that adding an additional arc to a DAG will always reduce expected delays. However, we find that adding arcs to the DAG may potentially increase, decrease, or not affect the average delays. This can be shown through the following two server example.

Consider the case of two independent $M/M/1$ queues. We will use $M_a$ to denote this menu. Let the arrivals rates be $\lambda_1^{(\epsilon)} = 1 - \epsilon\gamma_1$, and $\lambda_2^{(\epsilon)} = 1 - \epsilon\gamma_2$, and let $\mu_1 = \mu_2 = 1$. It is straightforward to calculate that $\widehat{W}_1 = 1/\gamma_1$ and $\widehat{W}_2 = 1/\gamma_2$. The average delay across both service classes is then

$$\bar{W}_a = \frac{1}{2}\left(\frac{1}{\gamma_1} + \frac{1}{\gamma_2}\right) \tag{2.19}$$

If we were to consider the alternative menu

$$M_b = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \tag{2.20}$$

then using Theorem 1 we find that $\widehat{W}_1 = 1/(\gamma_1 + \gamma_2)$ and $\widehat{W}_2 = 1/(\gamma_1 + \gamma_2) + 1/\gamma_2$. The average delay across both service classes is then

$$\bar{W}_b = \frac{1}{\gamma_1 + \gamma_2} + \frac{1}{2\gamma_2}. \tag{2.21}$$

Therefore the difference in average delays is

$$\Delta_{ab} := \bar{W}_b - \bar{W}_a = \frac{1}{\gamma_1 + \gamma_2} - \frac{1}{2\gamma_1}.$$

When $\gamma_1 = \gamma_2$, $\Delta_{ab} = 0$ and menus $M_a$ and $M_b$ have the same average delays. When $\gamma_1 > \gamma_2$, $\Delta_{ab}$ is positive, and menu $M_b$ has higher average delays than $M_a$, despite the additional flexibility. Otherwise, $\Delta_{ab}$ is negative, and menu $M_b$ has lower average delays than $M_a$.

This simple example demonstrates that adding additional flexibility to the design of the menu does not necessarily reduce the average delay (i.e., some form of Braess's paradox). Therefore if a service provider is considering adding additional flexibility to a system, it is important to carefully consider the way in which flexibility is being added.

**2.5.3 Numerical example** We will end this section by returning to our example in Figure 2.2 (a) to make some of the ideas discussed in the section more concrete. Recall the menu $M$ is given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2.22}$$

The limiting arrival rates are $\Lambda = (2, 1, 1, 2)$, and service rates are $\mu = (2, 1, 2, 1)$. We will let the sequence of arrival rates be $\lambda_i^{(\epsilon)} = \Lambda_i - \epsilon\gamma_i$ for $1 \leq i \leq 4$. We have three CRP components, $\mathbb{C}_1$ consisting of class 1 and server 1, $\mathbb{C}_2$ consisting of class 2 and server 2, and $\mathbb{C}_3$ consisting of classes 3 and 4 and servers 3 and 4.

We will begin by considering the question of implementability. We can see that the DAG induced by $M$ is a chained DAG, with $\mathbb{C}_1$ and $\mathbb{C}_2$ belonging to one partition in the chain, and $\mathbb{C}_3$ belonging to the other partition in the chain. Then Proposition 3 tells us that we can implement any waiting times $W_1 = W_2 > W_3 > 0$.

In this simple case, we can see which delays are implementable more directly, by looking at the exact expressions for the delays. Using Theorem 1, we can calculate the delays as

$$\hat{W}_1 = \frac{1}{\gamma_1} + \frac{1}{\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4}, \quad \hat{W}_2 = \frac{1}{\gamma_2} + \frac{1}{\gamma_1 + \gamma_2 + \gamma_3}, \quad \text{and } \hat{W}_3 = \frac{1}{\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4}.$$

By looking at these expressions, we can see that we can implement any delays $W_1$, $W_2$, and $W_3$ such that $W_3 > 0$, $W_1 > W_3$ and $W_2 > W_3$. To do this we would let $\gamma_1 = \frac{1}{W_1 - W_3}$,

$\gamma_2 = \frac{1}{W_2 - W_3}$, and $\gamma_3 + \gamma_4 = \gamma_1 + \gamma_2 - 1/W_1$.

This also suggests that in a congested system, a service provider is able to produce significant improvements in delay if they can make small changes to the arrival rates into the different service classes.

Suppose arrival rates are initially such that the slacks are proportional to arrival rates, i.e. $\gamma = \Lambda$, as in Afèche et al. (2021). The following table shows us the improvements in delay by adjusting the slacks so that $\gamma' = (9, 9, -3, -9)$ for different values of $\epsilon$. Note that $|\Lambda| = |\gamma'|$, so this adjustment does not alter the total arrival rate of customers into the system. We also show the percentage difference in average delays, denoted $\delta\bar{W}\%$, as well as the percentage of customers who are joining a different service class across the two scenarios, denoted $\delta\lambda\%$.

| $\epsilon$ | $\gamma$ | $\hat{W}_1$ | $\hat{W}_2$ | $\hat{W}_3$ | $\hat{W}_4$ | $\bar{W}$ | $\delta\bar{W}\%$ | $\delta\lambda\%$ |
|---|---|---|---|---|---|---|---|---|
| 0.1 | $\Lambda$ | 0.5727 | 1.0652 | 0.1649 | 0.1182 | 0.4353 | 61.43% | 33.33% |
| | $\gamma'$ | 0.2029 | 0.2001 | 0.2344 | 0.1237 | 0.1679 | | |
| 0.05 | $\Lambda$ | 0.6171 | 1.1151 | 0.1651 | 0.1408 | 0.4660 | 60.32% | 15.79% |
| | $\gamma'$ | 0.2351 | 0.2339 | 0.1830 | 0.1431 | 0.1849 | | |
| 0.01 | $\Lambda$ | 0.6563 | 1.1562 | 0.1662 | 0.1612 | 0.4929 | 56.82% | 3.03% |
| | $\gamma'$ | 0.2678 | 0.2677 | 0.1670 | 0.1613 | 0.2128 | | |

Table 2.1: Expected delays for different slacks.

As we can see, significant improvements in scaled delays are achieved while only changing the arrivals of a relatively small fraction of customers, with the improvements in comparison to the change required increasing as congestion increases.

Finally, we look at the question of menu design. In particular, we look at how we can change a menu to improve delays given a fixed CRP component structure, and fixed arrival rates. The residual matching for the menu $M$ in Equation (2.24) with limiting arrival rates

$\Lambda = (2, 1, 2, 1)$ and service rates $\mu = (2, 1, 1, 2)$ is

$$\breve{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.23}$$

There are 6 possible permutations of CRP components when the menu is just the residual matching $\breve{M}$, these permutations being all the permutations of the number (1, 2, 3). We can use Equation (2.17) to calculate the expected delay conditional on a particular permutation of CRP components. In this case, we will assume the slacks are $\gamma = (4, 3, 1, 1)$. The following table uses Equation (2.17) to calculate the conditional delays for all possible server permutations. We can see from this table that the permutation of CRP components that

| Permutation | Delay |
|---|---|
| (1,2,3) | $\frac{\mu_1}{\gamma_1} + \frac{\mu_1+\mu_2}{\gamma_1+\gamma_2} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 1.595$ |
| (1,3,2) | $\frac{\mu_1}{\gamma_1} + \frac{\mu_1+\mu_3+\mu_4}{\gamma_1+\gamma_3+\gamma_4} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 2$ |
| (2,1,3) | $\frac{\mu_2}{\gamma_2} + \frac{\mu_1+\mu_2}{\gamma_1+\gamma_2} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 1.429$ |
| (2,3,1) | $\frac{\mu_2}{\gamma_2} + \frac{\mu_2+\mu_3+\mu_4}{\gamma_2+\gamma_3+\gamma_4} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 8$ |
| (3,1,2) | $\frac{\mu_3+\mu_4}{\gamma_3+\gamma_4} + \frac{\mu_1+\mu_3+\mu_4}{\gamma_1+\gamma_3+\gamma_4} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 3$ |
| (3,2,1) | $\frac{\mu_3+\mu_4}{\gamma_3+\gamma_4} + \frac{\mu_2+\mu_3+\mu_4}{\gamma_2+\gamma_3+\gamma_4} + \frac{\mu_1+\mu_2+\mu_3+\mu_4}{\gamma_1+\gamma_2+\gamma_3+\gamma_4} = 2.967$ |

Table 2.2: Expected delays for different permutations of CRP components.

minimises delay is (2,1,3). We can then design a menu such that the DAG only admits this specific topological order. The DAG that achieves this is shown below.

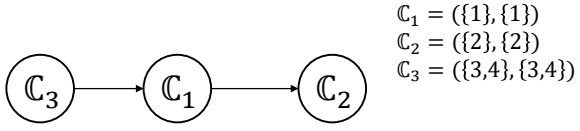



Figure 2.5: Delay minimising DAG.

This DAG can be achieved by having the service class in $\mathbb{C}_1$ served by the server in $\mathbb{C}_2$, and either of the service classes in $\mathbb{C}_3$ served by the server in $\mathbb{C}_1$. The following menu is one example of a menu that achieves this.

$$M' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.24}$$

In comparison, the original menu $M$ in Equation (2.24) with $\gamma = (4, 2, 1, 1)$ has average delays of 1.5, which as expected is higher than the average delays of our newly designed menu.

## 2.6 Proof of Main Results

We end this chapter by presenting proofs of our results.

**2.6.1 Proof of Theorem 1** The key observation needed to prove Theorem 1 is that only a relatively small subset of states have positive probability in heavy-traffic, and the information about which states have positive probability is captured by the CRP components and the DAG on the CRP components. However, before we go into more detail, it will be useful to introduce some notation. In section Equation (2.12), we defined the aggregate arrival rate for a CRP component $\mathbb{C}_k$ to be $\widetilde{\lambda}_k^{(\epsilon)} = \sum_{i \in \mathcal{C}_k} \lambda_i{}^{(\epsilon)} = \widetilde{\Lambda}_k - \epsilon\widetilde{\gamma}_k + o(\epsilon)$. For a subset of servers $S \subseteq [m]$, we define the *slack* for $S$ by:

$$\Delta(S) = \mu_S - \lambda_{U_S(M)}, \tag{2.25}$$

where $U_S(M)$ is defined in Proposition 1 as the subset of service classes that can only be served (or, uniquely served) by servers in $S$ under the menu $M$. For succinctness, we will suppress the dependence on $M$ in this section and use the notation $U(S)$ for $U_S(M)$.

It will also be useful to further aggregate the state space described in Section 2.2.1 so that the state depends only on the server permutation $s$ and the number of busy servers $b$, and not the number of customers. Specifically, for a server permutation $s = \{s_1, \ldots, s_m\}$ and $b \in \{0, 1, \ldots, m\}$ define:

$$P(s; b) = \{x \in X : x = (s_1, n_1, \ldots, s_b, n_b, s_{b+1}, s_{b+2} \ldots, s_m)\}$$

as the set of all states where $s$ is the ranking of servers in terms of the age of the customer for busy servers and the time since idleness for idle servers, and where exactly the first $b$ servers in $s$ are busy. We then have the following expression for the probability of the aggregate state $P(s; b)$:

$$\begin{aligned}\pi(P(s;b)) &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_b=0}^{\infty} \mathcal{B} \prod_{\ell=1}^{b} \frac{\lambda_{U(s_1,\ldots,s_\ell)}^{n_\ell}}{\mu_{\{s_1,\ldots,s_\ell\}}^{n_\ell+1}} \prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1} \\ &= \mathcal{B} \prod_{\ell=1}^{b} \frac{1}{\Delta(s_1, \ldots, s_\ell)} \prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1}. \end{aligned} \tag{2.26}$$

As a last step before developing the proof of Theorem 1, in Lemma 4 we state some properties of CRP components and topological orders that will be useful. This lemma has been slightly modified from (Afèche et al., 2021, Lemma 6).

**Lemma 4.** *Let $M$ be a service menu and $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ be its CRP components under a given heavy-traffic equilibrium strategy profile. For a CRP component $\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k)$ with non-empty $\mathcal{S}_k$ (i.e., $k \in [K']$) :*

(i) *The aggregate demand of service classes converges to the aggregate service rate as*

$\epsilon \to 0$, *that is,* $\widetilde{\Lambda}_k := \Lambda_{\mathcal{C}_k} = \mu_{\mathcal{S}_k} =: \widetilde{\mu}_k$ *(see* (2.12) *for definitions).*

(ii) *For any strict subset of servers* $S \subset \mathcal{S}_k$*, the set of service classes in residual matching* $\breve{M}$ *served only by* $S$ *is a strict subset of* $\mathcal{C}_k$*, and* $S$ *exhibits strictly positive slack as* $\epsilon \to 0$*, that is,*

$$\forall S \subset \mathcal{S}_k : U_S(\breve{M}) \subset \mathcal{C}_k \quad \textit{and} \quad \mu_S > \Lambda_{U_S(\breve{M})}.$$

*Further, since* $U_S(M) \subseteq U_S(\breve{M})$*, the positive slack condition also holds for* $U_S(M)$*.*

*(Recall that* $U_S(M)$ *is the subset of service classes that can only be served by servers in* $S$*.)*

*Let* $\sigma \in \mathcal{T}(\mathcal{D}, K')$ *be a topological order of the CRP components with non-empty server sets. Define* $\mathscr{S}_k = \mathcal{S}_{\sigma(1)} \cup \mathcal{S}_{\sigma(2)} \cup \cdots \cup \mathcal{S}_{\sigma(k)}$ *and* $\mathscr{C}_k = \mathcal{C}_{\sigma(1)} \cup \mathcal{C}_{\sigma(2)} \cup \cdots \cup \mathcal{C}_{\sigma(k)}$ *to be the subset of servers and service classes in the first* $k$ *CRP components in the topological order. Define*

$$\mathscr{C}'_k = \left\{ \cup_\kappa \mathcal{C}_\kappa | \kappa \in \{K'+1, \ldots, K\} : \exists k' \in \{1, \ldots, k\}, \kappa \in \mathsf{comps}(\sigma, k') \right\}$$

*to be the service classes of server-less CRP components that are part of* $\mathsf{comps}(\sigma, k')$ *for some* $k' \in [k]$*. Then,*

(iii) *Customers in* $\mathscr{C}_k \cup \mathscr{C}'_k$ *are exclusively served by servers in* $\mathscr{S}_k$*. That is,*

$$U_{\mathscr{S}_k}(M) = \mathscr{C}_k \cup \mathscr{C}'_k.$$

(iv) *The capacity slack of the set of servers* $\mathscr{S}_k$ *converges to zero as* $\epsilon \to 0$*, in particular,*

$$\Delta(\mathscr{S}_k) = \epsilon \sum_{\ell=1}^{k} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)} + o(\epsilon).$$

PROOF: See Appendix 2.D. □

We can now begin calculating the expected waits. Using the aggregated states from Equation (2.26), the following lemma (rephrased) from Afèche et al. (2021) gives an expression for the mean waiting time for each service class in terms of the probabilities $\pi(P(s;b))$.

**Lemma 5.** *(Afèche et al., 2021, Lemma 6) The steady-state mean waiting time of service class $i$ is equal to*

$$W_i = \sum_{s \in \Sigma_m} \sum_{b=1}^{m} W_i(s;b) \cdot \pi(P(s;b)),$$

*where $\Sigma_m$ denotes the set of all the permutations of $[m]$,*

$$W_i(s;b) = \sum_{\ell=1}^{b} \frac{\mathbb{1}\big(i \in U(s_1,\ldots,s_\ell)\big)}{\Delta(s_1,\ldots,s_\ell)},$$

*and $\pi(P(s;b))$ is given by* (2.26).

We are able to simplify these expressions further by showing that only a relatively small subset of aggregate states $(s,b)$ have asymptotically non-zero probabilities in heavy-traffic. These states are exactly those that are consistent with $\mathcal{T}(\mathcal{D}, K') = (\sigma_1,\ldots,\sigma_T)$ the collection of topological orders on $\{\mathbb{C}_1,\ldots,\mathbb{C}_{K'}\}$, a notion we will formalize in Definition 9. Our first step to showing this is to consider the slacks $\Delta(s_1,\ldots,s_\ell)$, which the preceding lemma suggests will be an important part of the analysis. Lemma 6 below, which is an extension of (Afèche et al., 2021, Lemma 4) shows that only certain subsets of servers have "interesting" slacks under a given sequence of arrival rates $\lambda^{(\epsilon)}$.

**Lemma 6.** *Let $\mathcal{D}$ be the DAG for the CRP decomposition $\{\mathbb{C}_1,\ldots,\mathbb{C}_{K'},\mathbb{C}_{K'+1},\ldots,\mathbb{C}_K\}$ under some menu $M$ and a given heavy-traffic equilibrium strategy profile. Then, a subset of servers $\{s_1,\ldots,s_\ell\} \subseteq [m]$ satisfies*

$$\lim_{\epsilon \to 0} \frac{\epsilon}{\Delta(s_1,\ldots,s_\ell)} > 0$$

*if and only if there exists a topological order* $\sigma \in \mathcal{T}(\mathcal{D}, K')$ *and an integer* $k$ *such that*

$$\{s_1, \ldots, s_\ell\} = \bigcup_{i=1}^{k} \mathcal{S}_{\sigma(i)}. \tag{2.27}$$

*Further, in this case :*

$$\lim_{\epsilon \to 0} \frac{\epsilon}{\Delta(s_1, \ldots, s_\ell)} = \frac{1}{\sum_{i=1}^{k} \widetilde{\gamma}_{\mathsf{comps}(\sigma,i)}}$$

*for any topological order* $\sigma$ *for which* (2.27) *is satisfied.*

PROOF: See Appendix 2.D. □

As implied in the previous paragraph, we can use Lemma 6 to prove Proposition 7 below, which states that a relatively small number of aggregate states have positive steady-state probability in heavy traffic; these are the aggregate states $P(s; m)$ in which $s$ is a permutation of the servers induced by a topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$ and such that all servers are busy.

**Definition 9.** (Server Permutations Induced by Topological Orders) *We say that a permutation of the servers* $s = (s_1, s_2, \ldots, s_m) \in \Sigma_m$ *is* induced by *the topological order* $\sigma \in \mathcal{T}(\mathcal{D}, K')$*, if* $s$ *can be expressed as a concatenation of sub-permutations:*

$$s = \left( \mathbf{s}_{\sigma(1)} || \mathbf{s}_{\sigma(2)} || \cdots || \mathbf{s}_{\sigma(K')} \right)$$

*with* $\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}$ *denoting a permutation of the servers* $\mathcal{S}_k$ *of CRP component* $\mathbb{C}_k$*. In other words, the servers of a CRP component are contiguous in the permutation* $s$*, and the order of the CRP components obeys the topological order* $\sigma$*.*

Returning to our four server example in Figure 2.3a, the CRP components were $\mathbb{C}_1 = (\mathcal{C}_1, \mathcal{S}_2 = (\{1\}, \{1\})$, $\mathbb{C}_2 = (\mathcal{C}_2, \mathcal{S}_2 = (\{2\}, \{2\})$, and $\mathbb{C}_3 = (\mathcal{C}_3, \mathcal{S}_3 = (\{3, 4\}, \{3, 4\})$, and the topological orders were $\sigma_a = (1, 2, 3)$ and $\sigma_b = (2, 1, 3)$. Definition 9 tells us the

topological order $\sigma_a$ induces two possible server permutations, $s_{a1} = (s_1||s_2||s_3||s_4)$ and $s_{a2} = (s_1||s_2||s_4||s_3)$.

The next proposition is an extension of (Afèche et al., 2021, Proposition 2).

**Proposition 7.** *Let $\mathcal{D}$ be the DAG for the CRP decomposition $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ under some menu $M$ and a heavy-traffic strategy profile. Let $s \in \Sigma_m$ be a server permutation.*

1. *If $b < m$, and/or $s$ is not a permutation of the servers induced by some topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$, then*

$$\lim_{\epsilon \to 0} \pi(P(s;b)) = 0.$$

2. *If $b = m$ and $s = \left(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')}\right)$ is a server permutation induced by topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$ with subpermutations $\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}$, then*

$$\lim_{\epsilon \to 0} \pi(P(s;b)) = \mathcal{B}' \cdot \mathbb{Q}(\sigma) \prod_{k=1}^{K'} \theta_k(\mathbf{s}_k)$$

*where $\mathcal{B}'$ is a normalization constant, $\mathbb{Q}(\sigma)$ was defined in (2.14) as*

$$\mathbb{Q}(\sigma) = \prod_{\kappa \in [K']} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}},$$

*and $\left\{\theta_k : \Sigma_{\mathcal{S}_k} \to \Re^+\right\}_{k \in [K']}$ is a fixed collection of functions mapping the sub-permutation of servers of CRP components to positive reals.*

Using Proposition 7 and the normalization condition $\sum_{s \in \Sigma_m, 0 \le b \le m} \pi(P(s;b)) = 1$, we

get:

$$\begin{aligned}\lim_{\epsilon\to 0}\sum_{s\in\Sigma_m,0\le b\le m}\pi(P(s;b)) &= \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')})\\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{k\in[K']}}}\pi(P(s;m))\\ &= \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')})\\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{k\in[K']}}}\mathcal{B}'\cdot\mathbb{Q}(\sigma)\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\\ &= \left(\sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\mathbb{Q}(\sigma)\right)\left(\mathcal{B}'\sum_{\{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{k\in[K']}}\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right),\end{aligned}$$

or,

$$\left(\mathcal{B}'\sum_{\{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{k\in[K']}}\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right)=\frac{1}{\sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\mathbb{Q}(\sigma)}.$$

Finally, we provide a lemma giving expressions for the scaled $W_i(s;b)$ when $s$ is a server permutation induced by a topological order $\sigma$, and $b=m$, as these are the only permutations that will be important in arriving at the result. A somewhat remarkable fact is that the limiting scaled $W_i(s;m)$ depends only on the topological order $\sigma$ and not the full server permutation $s$.

**Lemma 7.** *Let $s=(s_1,\ldots,s_m)$ be a server permutation induced by the topological order $\sigma\in\mathcal{T}(\mathcal{D},[K'])$. For a service class $i\in\mathbb{C}_k$,*

$$\lim_{\epsilon\to 0}\epsilon W_i(s;m)=w_{\sigma,k}\ :=\ \sum_{\kappa=\mathsf{comps}^{-1}(\sigma,k)}^{K'}\frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}. \tag{2.28}$$

Proof: See Appendix 2.D. □

Combining Proposition 7 with Lemmas 5-7, the limiting scaled mean waiting time for

service class $i \in \mathbb{C}_k$ is:

$$\begin{aligned}\widehat{W}_i^* &= \lim_{\epsilon \to 0} \epsilon \cdot W_i \\ &= \lim_{\epsilon \to 0} \sum_{s \in \Sigma_m} \epsilon \sum_{b=1}^{m} W_i(s;b) \cdot \pi(P(s;b)).\end{aligned}$$

Using the product rule of limits [1] we can reduce the above sum to a sum over server permutations induced by topological orders, and where all servers are busy.

$$\begin{aligned}\widehat{W}_i^* &= \lim_{\epsilon \to 0} \sum_{\sigma \in \mathcal{T}(\mathcal{D},K')} \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}\}_{k\in[K']}}} \epsilon \cdot W_i(s;m) \cdot \pi(P(s;m)) \\ &= \sum_{\sigma \in \mathcal{T}(\mathcal{D},K')} \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}\}_{k\in[K']}}} w_{\sigma,k} \cdot \mathcal{B}' \cdot \mathbb{Q}(\sigma) \prod_{\ell=1}^{K'} \theta_\ell(\mathbf{s}_\ell) \\ &= \sum_{\sigma \in \mathcal{T}(\mathcal{D},K')} w_{\sigma,k} \cdot \mathbb{Q}(\sigma) \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}\}_{k\in[K']}}} \mathcal{B}' \prod_{\ell=1}^{K'} \theta_\ell(\mathbf{s}_\ell) \\ &= \frac{\sum_{\sigma \in \mathcal{T}(\mathcal{D},[K'])} w_{\sigma,k} \cdot \mathbb{Q}(\sigma)}{\sum_{\sigma \in \mathcal{T}(\mathcal{D},K')} \mathbb{Q}(\sigma)} \\ &=: \widetilde{W}_k,\end{aligned}$$

as in the theorem statement.

**2.6.2 Proof of Theorem 2** Throughout this section, we will take the menu $M$, limiting arrival rates $\Lambda$ and service rates $\mu$, and slacks $\Gamma$ to be given, and largely suppress any dependence on $M$ in the notation. We will let $\breve{M}$ be the residual matching of the menu $M$ with arrival rates $\Lambda$ and service rates $\mu$.

1. Product rule of limits: If $\lim_{x\to x_0} f(x) = F$ and $\lim_{x\to x_0} g(x) = G$, then $\lim_{x\to x_0} f(x)g(x)$ exists and equals $FG$.

Instead of directly working with the matching rates $p_{ij}^{(\epsilon)}(M)$, we will look at the service probabilities $q_{ij}^{(\epsilon)}$. For all $i \in [n]$ and $j \in [m]$, $q_{ij}^{(\epsilon)}(x)$ is the probability with which server $j$ serves customer $i$ given the system is in state $x$ and server $j$ has become idle. We prove Theorem 2 by deriving and simplifying expressions for the limiting service probabilities $q_{ij}$ for the menu $M$, and find that the limiting service probabilities depend only on the service rates $\mu$, limiting arrival rates $\Lambda$, and the connectivity within each CRP component. To do this, we will make use of a new state space aggregation which we will introduce here.

In Section 2.6.1, we introduced the aggregate states $P(s, b)$ for ever $s \in \Sigma_m$ and $b \in [m]$. Recall that $P(s, b)$ is the set of all states where $s$ is the ranking of servers in terms of the age of the customers they are serving for busy servers, and the time since becoming idle for the idle servers, and $b$ is the number of busy servers. In this section, we further aggregate the state space, so that we can consider all of the states in which we observe a particular subpermutation of servers within a CRP component together. Specifically, for some $k \in [K']$ and some subpermutation $\mathbf{s}_k \in \Sigma_{\mathbf{S}_k}$, we define

$$P_k(\mathbf{s}_k) = \cup_{\sigma \in \mathcal{T}(\mathcal{D}, K')} \Big\{ s \in P(s,m) | s = \left( \mathbf{s}_{\sigma(1)} || \cdots || \mathbf{s}_k || \cdots || \mathbf{s}_{\sigma(K')} \right), \\ \mathbf{s}_\kappa \in \Sigma_{\mathbf{S}_\kappa} \text{ for } \kappa \in [K'] \text{ and } \kappa \neq k \Big\}$$

Note that while the set of aggregated states $P(s, b)$ does not depend on the menu being offered, $P_k(\mathbf{s}_k)$ depends on the set of topological orders, and hence does depend on the menu.

The first main step of our derivation will be to calculate the limiting service probabilities for our new further aggregated state space. That is, for each pair of service classes $i \in [n]$ and servers $j \in [m]$ in the same CRP component, and for any subpermutation of servers within that CRP component $\mathbf{s}_{k(k)} \in \Sigma_{\mathbf{S}_{k(j)}}$, we would like to calculate $q_{ij}(P_{k(j)}(\mathbf{s}_{k(j)}))$, the limiting service probability of service class $i$ by server $j$ given the system is in a state in

$P_{k(j)}(\mathbf{s}_{k(j)})$. Recall that $k(j)$ denotes the index of the CRP component that server $j$ belongs to. We do not consider $i$ and $j$ that are not in the same CRP component, as we know the limiting service probabilities of service classes and servers that are not in the same CRP component converge to zero. Similarly, we do not consider that service probabilities in any states $x$ not in $P_k(\mathbf{s}_k)$ for some $k \in [K']$ and $\mathbf{s}_k \in \Sigma_{\mathbf{S}_k}$, as those states have idle servers, and hence have probabilities converging to zero.

We will begin by writing the state dependent matching probability $q_{ij}^{(\epsilon)}(x)$ for an arbitrary state $x \in P_{k(j)}(\mathbf{s}_{k(j)})$. We will let $j(x)$ denote the position in the server permutation of server $j$ in the state $x$ and similarly will let $j(s)$ denote the position of server $j$ in the server permutation $s$. We can look at $q_{ij}^{(\epsilon)}(x)$ by conditioning on the position in the queuing network of the potential customer of type $i$ that $j$ serves. This lets us express $q_{ij(x)}^{(\epsilon)}$ as

$$\begin{aligned} q_{ij}^{(\epsilon)}(x) = &\sum_{r=j(x)}^{m} \left( \prod_{u=j(x)}^{r-1} \frac{\lambda^{n_u}_{\{U(s_1,\ldots,s_u \cap \overline{C(j)}\}}}{\lambda^{n_u}_{U(s_1,\ldots,s_u)}} \right) \left( \lambda_i \sum_{y=1}^{n_r} \frac{\lambda^{n_r-1}_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}}{\lambda^{n_r}_{U(s_1,\ldots,s_r)}} \right) \\ =&\lambda_i \sum_{r=j(x)}^{m} \left( \prod_{u=j(x)}^{r-1} \frac{\lambda^{n_u}_{\{U(s_1,\ldots,s_u)\cap\overline{C(j)}\}}}{\lambda^{n_u}_{U(s_1,\ldots,s_u)}} \right) \qquad (2.29) \\ &\times \left( \frac{\lambda^{n_r}_{U(s_1,\ldots,s_r)} - \lambda^{n_r}_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}}{\lambda^{n_r}_{U(s_1,\ldots,s_r)} \left( \lambda_{U(s_1,\ldots,s_r)} - \lambda_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}} \right)} \right). \qquad (2.30) \end{aligned}$$

It will be useful to decompose this expression into two parts, $q_{ij}^{+}(x)$, the part of the expression representing a transition within the CRP component, and $q_{ij}^{0}(x)$, the part of the expression representing a transition outside of the CRP component. We suppress the

dependence on $\epsilon$ to reduce clutter in the notation. So

$$q_{ij}^{+}(x) = \lambda_i \sum_{r=j(x)}^{m_k} \left( \prod_{u=j(x)}^{r-1} \frac{\lambda^{n_u}_{\{U(s_1,\ldots,s_u)\cap\overline{C(j)}\}}}{\lambda^{n_u}_{U(s_1,\ldots,s_u)}} \right) \times \left( \frac{\lambda^{n_r}_{U(s_1,\ldots,s_r)} - \lambda^{n_r}_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}}{\lambda^{n_r}_{U(s_1,\ldots,s_r)} \left(\lambda_{U(s_1,\ldots,s_r)} - \lambda_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}\right)} \right),$$

and $q_{ij}^{0}(x) = q_{ij}^{(\epsilon)}(x) - q_{ij}^{+}(x)$. Recall that $m_\kappa = \sum_{\ell\in[\kappa]} |\mathcal{S}_\ell|$, that is, $m_\kappa$ is the number of servers in the first $\kappa$ CRP components in the topological order.

As an intermediate step to looking at the aggregate matching probabilities $q_{ij}^{(\epsilon)}(P_k(\mathbf{s}_k))$, we will first look at the partially aggregated matching probabilities $q_{ij}^{(\epsilon)}(P(s,m))$.

$$q_{ij}^{(\epsilon)}(P(s,m)) = \frac{1}{\pi(P(s,m))} \left[ \sum_{x\in P(s,m)} \pi(x) q_{ij}^{+}(x) + \sum_{x\in P(s,m)} \pi(x) q_{ij}^{0}(x) \right].$$

However, the second term represents transitions from a state where the permutation of servers is induced by a topological order to a state where the permutation of servers is not induced by a topological order, and hence has a limiting probability of zero. This means we expect the second term in this expression to converge to zero, which we prove in the following lemma.

**Lemma 8.** *For a given admissible service menu $M$ with limiting arrival rates $\Lambda$, service rates $\mu$, and slacks $\Gamma$, let $\{\mathbb{C}_1,\ldots,\mathbb{C}_{K'},\mathbb{C}_{K'+1},\ldots,\mathbb{C}_K\}$ be the set of CRP components, and let $\mathcal{T}(\mathcal{D},K')$ be the set of topological orders on the CRP components. Then for any permutation of servers $s$ induced by some topological order $\sigma\in\mathcal{T}(\mathcal{D},K')$,*

$$\lim_{\epsilon\to 0} \sum_{x\in P(s,m)} \pi(x) q_{ij}^{0}(x) = 0$$

PROOF: See Appendix 2.E. □

We will now fix a topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$, and a server permutation $s \in \Sigma_m$ that is induced by $\sigma$. To reduce notational clutter, we assume without loss of generality that the CRP components are labelled in order of their position in the topological order, that is, $\sigma(k) = k$ for all $k \in K'$. Using Lemma 8, we can write $q_{ij}^{(\epsilon)}(P(s,m))$ as

$$q_{ij}^{(\epsilon)}(P(s,m)) = \frac{1}{\pi(P(s,m))} \sum_{x \in P(s,m)} \pi(x) q_{ij}^{+}(x) + o(1),$$

or written another way,

$$q_{ij}^{(\epsilon)}(P(s,m)) = \frac{\lambda_i}{\pi(P(s,m))} \sum_{n_1=0}^{\infty} \cdots \sum_{n_m=0}^{\infty} \mathcal{B} \prod_{\ell=1}^{m} \frac{\lambda_{U(s_1,\ldots,s_\ell)}^{n_\ell}}{\mu_{\{s_1,\ldots,s_\ell\}}^{n_\ell+1}} q_{ij}^{+}(s_1, n_1, \cdots, s_m, n_m) + o(1). \tag{2.31}$$

The following notation will be useful in simplifying this expression. Recall from Equation (2.25) that

$$\Delta(S) = \mu_S - \lambda_{U_S(M)}.$$

It will also be useful to define $\Delta_j(S)$ as

$$\Delta_j(S) = \mu_S - \lambda_{\{U_S(M) \cap \overline{C(j)}\}}. \tag{2.32}$$

We can then write Equation (2.31) as

$$q_{ij}^{(\epsilon)}(P(s,m)) = \frac{\mathcal{B}\lambda_i}{\pi(P(s,m))}\left(\prod_{\ell=m_{k(j)}+1}^{m}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\left(\prod_{\ell=1}^{m_{k(j)}-1}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)$$
$$\times\left(\prod_{\ell=m_{k(j)-1}+1}^{j-1}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\left[\sum_{r=j(s)}^{m_k(j)}\left(\prod_{u=j(s)}^{r}\frac{1}{\Delta_j(s_1,\ldots,s_u)}\right)\right.$$
$$\left.\times\left(\prod_{\ell=r+1}^{m_{k(j)}}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\left(\frac{1}{\Delta(s_1,\ldots,s_r)}-\frac{1}{\Delta_j(s_1,\ldots,s_r)}\right)\right]+o(1), \quad (2.33)$$

where as before $m_\kappa = \sum_{\ell\in[\kappa]}|\mathcal{S}_\ell|$.That is, $m_\kappa$ is the number of servers in the first $\kappa$ CRP components in the topological order.

We saw in Section 2.6.1 that the limiting values of $\Delta(s_1,\ldots,s_\ell)$ depend on the values of $\ell$. If $\ell = m_\kappa$ for some $\kappa \in [K']$, then we know from Lemma 6 that

$$\lim_{\epsilon\to 0}\frac{\epsilon}{\Delta(s_1,\ldots,s_{m_\kappa})} = \frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}.$$

For all other values of $\ell$, there is some $\kappa \in [K']$ such that $m_{\kappa-1}+1 \le \ell \le m_\kappa - 1$. Here we take $m_0 = 0$. We let $S = \{s_{m_{\kappa-1}+1},\ldots,s_\ell\}$. Following the outline in (Afèche et al., 2021, Lemmas 5 and 8), we can show that:

$$\lim_{\epsilon\to 0}\Delta(s_1,\ldots,s_\ell) = \mu_S - \Lambda_{U_S(\breve{M})} > 0.$$

In particular, this means that for all $\kappa \in [K']$, and $m_{\kappa-1}+1 \le \ell \le m_\kappa - 1$, $\lim_{\epsilon\to 0}\Delta(s_1,\ldots,s_\ell)$ is a real number greater than zero that depends only on the permutation of servers in $\mathbb{C}_\kappa$.

The same reasoning implies that for all $j \le \ell \le m_{k(j)}$, $\lim_{\epsilon\to 0}\Delta_j(s_1,\ldots,s_\ell)$ is a real number greater than zero that depends only on the permutation or servers in $\mathbb{C}_k$.

We can use these observations to prove the following lemma.

**Lemma 9.** *We can find functions* $\left\{\theta_\kappa : \Sigma_{\mathcal{S}_\kappa} \to \Re^+\right\}_{\kappa\in[K']}$, $H_{ij} : \Sigma_{\mathcal{S}_{k(j)}} \to \Re^+$, *and* $G_{ij} : \Sigma_{\mathcal{S}_{k(j)}} \to \Re^+$, *such that* $q_{ij}(P(s,m)) = \lim_{\epsilon\to 0} q_{ij}^{(\epsilon)}(P(s,m))$ *can be written as*

$$q_{ij}(P(s,m)) = \lim_{\epsilon\to 0}\left[\frac{\mathcal{B}\lambda_i}{\pi(P(s,m))\epsilon^{K'}}\mathbb{Q}(\sigma)\left(\prod_{\kappa\neq k(j)}\theta_\kappa(s_\kappa)\right)H_{ij}(\mathbf{s}_{k(j)})\right]$$
$$-\lim_{\epsilon\to 0}\left[\frac{\mathcal{B}\lambda_i}{\pi(P(s,m))\epsilon^{K'-1}}\left(\prod_{\kappa\neq k}\frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}\right)\right. \tag{2.34}$$
$$\left.\left(\prod_{\kappa\neq k(j)}\theta_\kappa(s_\kappa)\right)G_{ij}(\mathbf{s}_{k(j)}) + o(1)\right], \tag{2.35}$$

*where* $\theta_\kappa$ *and* $H_{ij}$ *only depend on* $\breve{M}$, $\Lambda$, *and* $\mu$.

PROOF: See Appendix 2.E. □

We provide exact definitions of $\left\{\theta_k : \Sigma_{\mathcal{S}_k} \to \Re^+\right\}_{k\in[K']}$, $H_{ij} : \Sigma_{\mathcal{S}_k} \to \Re^+$, and $G_{ij} : \Sigma_{\mathcal{S}_k} \to \Re^+$ in the proof of Lemma 9 in Appendix 2.E.

Notice that the first line in Equation (2.34) has an $\epsilon^{-K'}$ term, and the second line has an $\epsilon^{-(K'-1)}$ term. Since $q_{ij}$ are probabilities and therefore must be between 0 and 1, we know that $\lim_{\epsilon\to 0} B\epsilon^{-K'}$ is bounded. This implies that $\lim_{\epsilon\to 0} B\epsilon^{-(K'-1)} = 0$, and so only the first line in Equation (2.34) will be non-zero. Thus

$$q_{ij}(P(s,m)) = \frac{\mathcal{B}'\lambda_i}{\pi(P(s,m))}\mathbb{Q}(\sigma)\left(\prod_{\kappa\neq k}\theta_\kappa(s_\kappa)\right)H_{ij}(\mathbf{s}_{k(j)}). \tag{2.36}$$

Because $q_{ij}$ are matching probabilities, we also know that

$$q_{ij}(P(s,m)) = \frac{q_{ij}(P(s,m))}{\sum_{i'\in\mathcal{C}_{k(j)}} q_{i'j}(P(s,m))}. \tag{2.37}$$

Since the only term in Equation (2.37) that depend on $j$ is the $H_{ij}(\mathbf{s}_{k(j)})$ term, we can write

$q_{ij}(P(s,m))$ as

$$q_{ij}(P(s,m)) = \frac{H_{ij}(\mathbf{s}_{k(j)})}{\sum_{i' \in \mathcal{C}_k} H_{i'j}(\mathbf{s}_{k(j)})}. \tag{2.38}$$

Since Equation (2.38) holds for any server permutation $s \in \Sigma$, and depends only on $\mathbf{s}_k$ and not on the rest of the server permutation, this implies that

$$q_{ij}(P_{k(j)}(\mathbf{s}_{k(j)})) = \frac{H_{ij}(\mathbf{s}_{k(j)})}{\sum_{i' \in \mathcal{C}_k} H_{i'j}(\mathbf{s}_{k(j)})}. \tag{2.39}$$

Since, as Lemma 9 states, $H_{ij}(\mathbf{s}_{k(j)})$ does not depend on $\Gamma$, the remaining step needed to prove Theorem 2 is to show that $\pi(P_{k(j)}(\mathbf{s}_{k(j)}))$ also does not depend on $\Gamma$. This is captured in the following lemma.

**Lemma 10.** *For an admissible service menu $M$ with limiting arrival rates $\Lambda$ service rates $\mu$, and slacks $\Gamma$, the limiting probability of being in a state with the sub-permutation of server $\mathbf{s}_k \in \Sigma_{\mathbf{S}_k}$ for $k \in K'$ is equal to*

$$\lim_{\epsilon \to 0} \pi(P_k(\mathbf{s}_k)) = \frac{\theta_k(s_k)}{\sum_{\mathbf{s}_\kappa \in \Sigma_{\mathcal{S}_k}} \theta_\kappa(\mathbf{s}_\kappa)},$$

*where $\left\{\theta_\kappa : \Sigma_{\mathcal{S}_\kappa} \to \Re^+\right\}_{\kappa \in [K']}$ is a function that depends only on $\breve{M}$, $\Lambda$, and $\mu$.*

PROOF: See Appendix 2.E. □

Combining Lemma 10 with Equation (2.38), we have that the limiting service probabilities $\lim_{\epsilon \to 0} q_{ij}^{(\epsilon)}$ do not depend on the exact values of the slacks $\Gamma$, only requiring that $M$ is an admissible menu for the slacks $\Gamma$.

## 2.A Section 2.3 Proofs

PROOF OF LEMMA 1: Let us define the set $\mathcal{F}_{\max}$ as

$$\mathcal{F}_{\max} := \left\{ \sum_{i\in[n]} f = [f_{ij}] : \quad \sum_{i\in[n]} f_{ij} \le \mu_j \quad \forall j \in [m] \quad , f \ge 0, \quad f_{ij} = 0, \quad \forall (i,j) : m_{ij} = 0 \right\}.$$

Note that for all $\epsilon \in [0, \epsilon_o)$, $\mathcal{F}(\epsilon, \lambda^{(\epsilon)}, M) \subseteq \mathcal{F}_{\max}$. Furthermore, since $\mathcal{F}_{\max}$ is a compact set, we know that the sequence $f^{(\epsilon)}$ has a subsequence that converges to some limit in $\mathcal{F}_{\max}$. Let $\tilde{f}$ denote this limit. To prove that $\tilde{f} \in \mathcal{F}(0, \lambda^{(\epsilon)}, M)$, all that remains to be shown is that $\tilde{f}$ satisfies

$$\sum_{j\in[m]} \tilde{f}_{ij} = \Lambda_i, \quad \text{for all } i \in [n].$$

But we know that

$$\sum_{j\in[m]} f_{ij}^{(\epsilon)} = \lambda_i^{(\epsilon)}, \quad \text{for all } i \in [n] \text{ and } 0 < \epsilon < \epsilon_0,$$

and $\tilde{f}$ is the limit of a subsequence of $f^{(\epsilon)}$, and so

$$\sum_{j\in[m]} \tilde{f}_{ij} = \lim_{\epsilon\to 0} \lambda_i^{(\epsilon)} = \Lambda_i, \quad \text{for all } i \in [n]$$

as required. □

PROOF OF LEMMA 2: Fix a topological order $\sigma_t \in \mathcal{T}(\mathcal{D}, [K'])$ and an index $\kappa \in [K']$. Define the sets

$$\mathscr{C} = \bigcup_{\ell=1}^{\kappa} \{\mathcal{C}_i : i \in \mathsf{comps}(\sigma_t, \ell)\}, \quad \text{and} \quad \mathscr{S} = \bigcup_{\ell=1}^{\kappa} \{\mathcal{S}_i : i \in \mathsf{comps}(\sigma_t, \ell)\}.$$

By the definition of the DAG $\mathcal{D}$ and topological order $\sigma_t$, we have that

$$\mathscr{S} = S(\mathscr{C}).$$

That is, the services classes $\mathscr{C}$ are only served by servers in $\mathscr{S}$. We can find a lower bound on the scaled mean waiting times of the service classes in $\mathscr{C}$ using the scaled mean waiting time of a $M/M/1$ queue:

$$\sum_{i\in\mathscr{C}} \lambda_i^{(\epsilon)} \widehat{W}_i^{(\epsilon)} \geq \frac{\epsilon}{\mu_{\mathscr{S}} - \lambda_{\mathscr{C}}^{(\epsilon)}}. \tag{2.A1}$$

Further, from Lemma 4 we know that,

$$\mu_{\mathscr{S}} - \lambda_{\mathscr{C}}^{(\epsilon)} = \epsilon \sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)} + o(\epsilon).$$

If, contradictory to the Lemma 2, $\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)} \leq 0$, then the right-hand side of (2.A1) must diverge, and hence the sum on the left-hand side as well. However, from the admissibility of $M$, each $\widehat{W}_i^{(\epsilon)}$ converges, and therefore also the sum on the left-hand side of (2.A1). Thus we must have $\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)} > 0$ for all $\sigma_t \in \mathcal{T}(\mathcal{D}, [K'])$ and $\kappa \in [K']$. □

## 2.B Section 2.4 Proofs

Proof of Proposition 2: Given $M$ is admissible for $(\lambda^{(\epsilon)}, \mu)$, we know from the definition of admissibility that

$$\Delta_{\mathscr{S}}^{(\epsilon)}(M) := \sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \lambda_i^{(\epsilon)} = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m]. \tag{2.A1}$$

To show $M$ is admissible for $(\Lambda - \epsilon\Lambda, \mu)$, we must show that

$$\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i + \epsilon \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m]. \tag{2.A2}$$

Equation (2.A1) implies that $\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i \geq 0$ for all $\mathscr{S} \subseteq [m]$. For any $\mathscr{S} \subseteq [m]$ such that $\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i > 0$, Equation (2.A2) holds without regardless of the $\epsilon$ terms. In the case that $\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i = 0$, then

$$\begin{aligned}\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i + \epsilon \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i &= \epsilon \sum_{i\in U_{\mathscr{S}}(M)} \Lambda_i \\ &= \epsilon \sum_{j\in\mathscr{S}} \mu_j.\end{aligned}$$

But $\sum_{j\in\mathscr{S}} \mu_j > 0$, so $\epsilon \sum_{j\in\mathscr{S}} \mu_j = \Omega(\epsilon)$ as required.

The second part of the proposition states that $\breve{M}$ is admissible for $(\Lambda - \epsilon\Lambda, \mu)$. To show this, similarly to the first part of the proposition we must show that

$$\sum_{j\in\mathscr{S}} \mu_j - \sum_{i\in U_{\mathscr{S}}(\breve{M})} \Lambda_i + \epsilon \sum_{i\in U_{\mathscr{S}}(\breve{M})} \Lambda_i = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m]. \tag{2.A3}$$

There are two cases to consider. In the first case, $\mathscr{S} = \cup_{k\in T}\mathcal{S}_k$ for some $T \subseteq K$. In words, this means that $\mathscr{S}$ is the union of servers in a particular subset of CRP components. It is shown in the proof of (Afèche et al., 2021, Lemma 4) that in this case, $\sum_{j\in\mathscr{S}} \mu_j = \sum_{i\in U_{\mathscr{S}}(\breve{M})} \Lambda_i$, and Equation (2.A3) holds following the same reasoning as in the first part of the proposition. In the second case, $\mathscr{S} \neq \cup_{k\in T}\mathcal{S}_k$ for any $T \subseteq K$, and the proof of (Afèche et al., 2021, Lemma 4) shows that $\sum_{j\in\mathscr{S}} \mu_j > \sum_{i\in U_{\mathscr{S}}(\breve{M})} \Lambda_i$, and Equation (2.A3) holds following similar reasoning as in the first part of the proposition. □

## 2.C Section 2.5 Proofs

PROOF OF COROLLARY 2: We will prove this corollary by proving the contrapositive. So suppose there are $k \in [K]$ and $\kappa \in [K]$ such that there are no topological orders $\sigma \in \mathcal{T}(\mathcal{D}, K')$ with $\mathsf{comps}^{-1}(\sigma, \kappa) \leq \mathsf{comps}^{-1}(\sigma, k)$. This means that in every topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$, $\mathsf{comps}^{-1}(\sigma, \kappa) > \mathsf{comps}^{-1}(\sigma, k)$. From the definition of the conditional delay $w_{\sigma,k}$ in Equation (2.15), this implies that $w_{\sigma,k} > w_{\sigma,\kappa}$ for all $\sigma \in \mathcal{T}(\mathcal{D}, K')$. As the total delays are weighted sums of the conditional delays, this proves the result. □

PROOF OF PROPOSITION 3: Without loss of generality let us index the CRP components in such a way that $W_k \leq W_{k+1}$ for all $k \in [K-1]$. Recall $\{\mathscr{C}_1, \ldots, \mathscr{C}_L\}$ is the partition described in Definition 7. Are stated in the proposition, we will assume

(i) $W_k = W_\kappa$ for all $(k, \kappa) \in [K] \times [K]$ such that $W_k \in \mathscr{C}_\ell$ and $W_\kappa \in \mathscr{C}_\ell$ for some $\ell \in [L]$,

(ii) $W_k < W_\kappa$ for all $(k, \kappa) \in [K] \times [K]$ such that $W_k \in \mathscr{C}_\ell$ and $W_\kappa \in \mathscr{C}_{\ell'}$ for some $(\ell, \ell') \in [L] \times [L]$ where $\ell < \ell'$.

We will now show how to choose a vector of capacity slacks $\tilde{\gamma} = (\tilde{\gamma}_1, \ldots, \tilde{\gamma}_K)$ such that $W_{\mathbb{C}_k} = W_k$ for all $k \in [K]$. Fix $\tilde{\gamma}$ such that $\widetilde{\gamma}_k = \widehat{\gamma}_\ell$ for all $k \in \mathscr{C}_\ell$. It follows from the chained structure of the DAG and the construction of $\tilde{\gamma}$ that for any permutation $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(K))$ induced by some topological order the vector $(\widetilde{\gamma}_{\sigma^{-1}(1)}, \widetilde{\gamma}_{\sigma^{-1}(2)}, \ldots, \widetilde{\gamma}_{\sigma^{-1}(K)})$ is constant. This observation together with Theorem 1 imply that $\mathbb{Q}(\sigma)$ in Equation (2.14) is also constant, independent of $\sigma$. Furthermore, by symmetry, it is not hard to see that two CRP components that belong to the same partition $\mathscr{C}_\ell$ have the same limiting scaled waiting times, which we denote by $\widehat{\mathbb{W}}_\ell$. One can show from Theorem 1 that

$$\widehat{\mathbb{W}}_\ell = \widehat{\mathbb{W}}_{\ell-1} + \frac{1}{n_\ell} \sum_{s=1}^{n_\ell} \frac{1}{\sum_{j=\ell+1}^{L} n_j \, \widehat{\gamma}_j + s \, \widehat{\gamma}_\ell}, \qquad \ell = 1, 2 \ldots, L \tag{2.A1}$$

with $\widehat{\mathbb{W}}_0 = 0$. We use this condition to find the values of $\{\widehat{\gamma}_\ell\}$ that implement $\{\mathbb{W}_\ell\}$, that is, $\widehat{\mathbb{W}}_\ell = \mathbb{W}_\ell$ for all $\ell \in [L]$. To this end, we use backward induction on $\ell$. For $\ell = L$ we have that

$$\widehat{\mathbb{W}}_L = \widehat{\mathbb{W}}_{L-1} + \frac{1}{n_L} \sum_{s=1}^{n_L} \frac{1}{s\,\widehat{\gamma}_L}.$$

Thus, $\widehat{\gamma}_L$ must satisfy

$$\widehat{\gamma}_L = \frac{1}{(\mathbb{W}_L - \mathbb{W}_{L-1})} \frac{1}{n_L} \sum_{s=1}^{n_L} \frac{1}{s}.$$

Now suppose that we have determined the values of $\widehat{\gamma}_L, \widehat{\gamma}_{L-1}, \ldots, \widehat{\gamma}_{\ell+1}$ and define $\widehat{\Gamma}_\ell := \sum_{j=\ell+1}^{L} n_j\,\widehat{\gamma}_j$. We find the value $\widehat{\gamma}_\ell$ by solving (2.A1)

$$\mathbb{W}_\ell = \mathbb{W}_{\ell-1} + \frac{1}{n_\ell} \sum_{s=1}^{n_\ell} \frac{1}{\widehat{\Gamma}_\ell + s\,\widehat{\gamma}_\ell}.$$

We note that there exists a unique $\widehat{\gamma}_\ell$ that solves this equation in the region $\widehat{\gamma}_\ell > -\widehat{\Gamma}_\ell/n_\ell$. This follows from the fact that the summation above is monotonically decreasing in $\widehat{\gamma}_\ell$ in this region and diverges to $+\infty$ as $\widehat{\gamma}_\ell$ approaches $\widehat{\Gamma}_\ell/n_\ell$ from above and converges to zero as $\widehat{\gamma}_\ell$ approaches $\infty$. □

Proof of Proposition 4: Note from (2.14) that

$$w_{\sigma,k} := \sum_{\kappa=\sigma^{-1}(k)}^{K} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\sigma(\ell)}} = \frac{1}{|a|} + \sum_{\kappa=\sigma^{-1}(k)}^{K-1} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\sigma(\ell)}}.$$

Let us prove that $w_{\sigma,k} \geq 1/|a|$. From the previous equation, this would follow if the last summation is nonnegative. Suppose, by contradiction that this is not the case. Then, there exists a $\kappa$ such that $\sigma^{-1}(k) \leq \kappa \leq K-1$ such that $\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\sigma(\ell)} < 0$. In other words, the cumulative capacity slack of the CRP components $\{\mathbb{C}_{\sigma(1)}, \mathbb{C}_{\sigma(2)}, \ldots, \mathbb{C}_{\sigma(\kappa)}\}$ is negative. However, this would imply that the cumulative arrival rate to these components exceeds

the total service capacity of all the servers in these components. This, together with the DAG structure connecting all the CRP components imply that the stability condition in Proposition 1 is violated, which holds by assumption. From this contradiction we conclude that $w_{\sigma,k} \geq 1/|a|$ and then from (2.16) we also get that $W_{\mathbb{C}_k} \geq 1/|a|$.

Let us now prove the second part of the corollary, namely, there can be at most one CRP component $\hat{\kappa} \in [K]$ such that $\widehat{W}_{\mathbb{C}_{\hat{\kappa}}} = 1/|a|$. From the previous discussion, it follows that the requirement $\widehat{W}_{\mathbb{C}_{\hat{\kappa}}} = 1/|a|$ can only be satisfied if $w_{\sigma,\hat{\kappa}} = 1/|a|$ for all permutations $\sigma$ associated a topological order. But this can only happen if $\sigma^{-1}(\hat{\kappa}) = K$ for all permutation $\sigma$. Evidently, this condition can only be satisfied by at most one CRP component and holds trivially if $K = 1$. $\square$

Proof of Proposition 5: Take any slacks $\gamma$ with $|\gamma| > 0$. We will first show that $M$ is admissible with $\lambda^{(\epsilon)} = \Lambda - \epsilon\gamma + o(\epsilon)$ and $\mu$. To do this, we need to show that

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m],$$

where

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\mathscr{S}}(M)} \lambda^{(\epsilon)}_i.$$

We define $D_{\mathscr{S}}$ as

$$D_{\mathscr{S}} = \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\mathscr{S}}(M)} \Lambda_i$$

for all $\mathscr{S} \subseteq [m]$. Then

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) = D_{\mathscr{S}} + \epsilon \sum_{j \in \mathscr{S}} \gamma_i + o(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m],$$

From the definition of $M$ we know that $D_{\mathscr{S}} > 0$ for all $\mathscr{S} \subseteq [m]$, implying that $\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega\epsilon$

for all $\mathscr{S} \subseteq [m]$. For the case of $\mathscr{S} = [m]$, since $|\Lambda| = |\mu|$, and $|\gamma| > 0$,

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) = +\epsilon|\gamma| + o(\epsilon) = \Omega(\epsilon)$$

as required.

What remains to be shown is that $M$ induces a single CRP component. This follows from part (i) of Lemma 4, which states that within a CRP component $\widetilde{\Lambda}_k := \Lambda_{\mathcal{C}_k} = \mu_{\mathcal{S}_k} =: \widetilde{\mu}_k$ (see (2.12) for definitions). But with our choice of $M$, we know that for any subset of servers $\mathscr{S} \subsetneq [m]$, any subset of customers classes $\mathscr{C} \subseteq [n]$ such that every class in $\mathscr{C}$ is compatible with some server in $\mathscr{S}$ will have $\Lambda_{\mathscr{C}} < \mu_{\mathscr{S}}$. Thus there are no CRP components that do not consist of all service classes and all servers, implying there is exactly one CRP component.

□

Proof of Lemma 3: We assume without loss of generality that the CRP components are labelled so that $\mathsf{comps}^{-1}(\sigma, k) = k$ for all $k \in [K']$. We construct the menu $M$ as follows. Let $\breve{M}$ be any residual matching associated with the collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$. Construct the menu $M$ as follows. Let $m_{ij} = 1$ for all $i \in [n]$ and $j \in [m]$ such that $\breve{m}_{ij} = 1$. Then for every $k \in [K' - 1]$, let $m_{ij} = 1$ for some $i \in \mathcal{C}_{k+1}$ and some $j \in \mathcal{S}_k$. That is, for every CRP component $\mathbb{C}_k$ for $k \in [K' - 1]$, we assign some service class in $\mathbb{C}_{k+1}$ to be a served by a server in $\mathbb{C}_k$. We will show that this has the effect of adding an arc to the DAG from $\mathbb{C}_{k+1}$ to $\mathbb{C}$ without altering the CRP component structure.

We will begin by assuming that there are no service classes with zero arrivals, that is, we assume that $\tilde{\Lambda}_k > 0$ for all $k \in [K]$, and $K = K'$. In this case, we let $m_{ij} = 0$ for all other combinations of $i \in [n]$ and $j \in [m]$. We will mention at the end of this proof how to adjust the menu $M$ for the case in which there is at least one $k \in [K]$ with $\tilde{\Lambda}_k = 0$.

The next step is to show that the CRP components of $M$ are $\mathbb{C}$. This is equivalent

to showing that $\mathcal{F}(0,\Lambda,M) = \mathcal{F}(0,\Lambda,\breve{M})$. First note that there can only be flow between servers in $\mathcal{S}_k$ and customers in $\mathcal{C}_k \cup \mathcal{C}_{k+1}$ for $k \in [K-1]$, and there can only be flow between servers in $\mathcal{S}_K$ and customers in $\mathcal{C}_K$ due to the construction of $M$. But there can be no flow between servers in $\mathcal{S}_1$ and customers in $\mathcal{C}_2$, as all of the capacity of servers in $\mathcal{S}_1$ needs to be allocated to servers in $\mathcal{C}_1$, since $\tilde{\Lambda}_1 = \tilde{\mu}_1$. It can then be argued inductively that servers in $\mathcal{S}_k$ do not have the capacity to allocate flow to customers in $\mathcal{C}_{k+1}$, even though there is a server that has the compatibility to do so. Thus $\mathcal{F}(0,\Lambda,M) = \mathcal{F}(0,\Lambda,\breve{M})$ as required.

Next, we will show that the DAG of $M$ only admits the topological order $\sigma$. This is true based on the construction of $M$. The only arcs in $M$ that are not in the residual matching $\breve{M}$ are between components $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$ for $k \in [K-1']$, and there is such an arc for $k \in [K-1]$. Thus we require for any topological order $\sigma_t$ admitted by $M$ that $\sigma_t(k) < \sigma_t(k+1)$ for $k \in [K-1]$. But the only topological order that achieves this is $\sigma$, where as stated previously $\sigma(k) = k$.

The final step needed to prove the first claim in Lemma 3 is to show that $M$ is admissible. Recall from Definition 1 that for a menu to be admissible we require that $\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega(\epsilon)$ for all $\mathcal{S} \subseteq [m]$, where

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\mathscr{S}}(M)} \lambda^{(\epsilon)}_i.$$

The proof of (Afèche et al., 2021, Lemma 4) argues that if the subset of servers $\mathscr{S} \subseteq [m]$ is not equal to $\cup_{\ell=1}^{k} \mathcal{S}_\ell$ for some $k \in [K]$, then

$$\mu_S - \Lambda_{U_S(M)} > 0.$$

which means that $\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega(\epsilon)$ for all $\mathscr{S} \subseteq [m]$ that is not equal to $\cup_{\ell=1}^{k} \mathcal{S}_\ell$ for some $k \in [K]$. For $\mathscr{S} \subseteq [m]$ such that $\mathscr{S} = \cup_{\kappa=1}^{k} \mathcal{S}_\ell$ for some $k \in [K]$, we know from Lemma 4

that $\sum_{j\in\mathscr{S}}\mu_j = \sum_{i\in U_{\mathscr{S}}(M)}\Lambda_i$. So

$$\Delta^{(\epsilon)}_{\mathscr{S}}(M)\sum_{\ell=1}^{k}\epsilon\tilde{\gamma}_\ell - o(\epsilon).$$

But since from the statement of the lemma, $\sum_{\ell=1}^{k}\epsilon\tilde{\gamma}_\ell > 0$ for all $k\in[K]$, this means that $\Delta^{(\epsilon)}_{\mathscr{S}}(M) = \Omega(\epsilon)$ as required. Hence $M$ is admissible as claimed.

This also demonstrates why no admissible menu $M$ can admit a topological order $\sigma$ such that $\sum_{\ell=1}^{k}\epsilon\tilde{\gamma}_\ell \leq 0$ for some $k\in[K']$. If that were the case, then we would have that $\lim_{\epsilon\to 0}\Delta^{(\epsilon)}_{\mathscr{S}}(M)\leq 0$ for $\mathscr{S} = \cup_{\kappa=1}^{k}\mathcal{S}_\ell$, which contradicts $M$ being admissible. This holds even if we were to consider the scenario in which $\tilde{\Lambda}_k = 0$ for some $k\in[K]$, as this would only decrease the values of $\tilde{\gamma}_{\mathsf{comps}(\sigma),k}$, making it more difficult to satisfy the condition $\lim_{\epsilon\to 0}\Delta^{(\epsilon)}_{\mathscr{S}}(M) > 0$.

Finally, we will mention how we can extend the construction of $M$ to account for CRP components $k$ with $\tilde{\Lambda}_k = 0$. Recall that these CRP components do not influence the topological orders themselves, only the slacks the elements $\mathsf{comps}(\sigma,k)$. We require for the admissibility of $M$ that $\sum_{\ell=1}^{k}\mathsf{comps}(\sigma,\ell) > 0$ for all $k\in[K']$. This can potentially be achieved in many ways, one of which will always be to let $m_{ij} = 1$ for some $j$ in $\mathbb{C}_{K'}$ and for all $i\in[n]$ such that $\Lambda_i = 0$. This construction will mean that $\tilde{\gamma}_{\mathsf{comps}(\sigma,k)} = \tilde{\gamma}_k$ for all $k\in[K'-1]$, and $\tilde{\gamma}_{\mathsf{comps}(\sigma,K')} = \tilde{\gamma}_{K'} + \sum_{i:\Lambda_i=0}\gamma_i$. Thus $\sum_{\ell=1}^{k}\mathsf{comps}\sigma,\ell = \sum_{\ell=1}^{k}\tilde{\gamma}_k > 0$ for all $k\in[K'-1]$, and $\sum_{\ell=1}^{K'}\mathsf{comps}\sigma,\ell = |\gamma| > 0$ as required. □

Proof of Proposition 6: Because the total delays are weighted averages of conditional delays, we know if the only conditional delay we are taking the average over is the minimum possible conditional delay, we will achieve the minimum total delay. From Lemma 3, we know for any admissible menu $M$, the only topological orders with positive probability are those that are admissible.

Because the set of all permutations of CRP components is finite, the set of admissible

topological orders is finite. Thus there will be some implementable topological order that achieves the minimum conditional delay (If there are some $i \in [n]$ such that $\Lambda_i = 0$, for each topological order we would also need to consider the assignment of customers classes with zero arrivals to servers that minimises delay for each topological order).

Therefore we will be able to minimise the total average delay by choosing an admissible menu $M$ that only allows for the admissible topological order that achieves the minimum conditional delay. We know that such a menu exists from Lemma 3. □

## 2.D Section 2.6.1 Proofs

Proof of Lemma 4: There are two differences between the setup in our paper and in Afèche et al. (2021): first, the constants $\gamma_i$ for the approach to heavy-traffic are allowed to be arbitrary, while in Afèche et al. (2021) the authors impose $\gamma_i = \Lambda_i$. Second, our setup has service classes with $\Lambda_i = 0$ and hence CRP components which consist of a single service class and no servers. Despite these, the proofs for parts (i) and (ii) are identical to the proofs of parts (i) and (ii) of (Afèche et al., 2021, Lemma 3).

Part (iii) of (Afèche et al., 2021, Lemma 3) states that $U_{\mathscr{S}_k}(M) = \mathscr{C}_k$, which in our setup should be interpreted as

$$U_{\mathscr{S}_k}(M) \cap \left\{ \cup_{\ell=1}^{K'} \mathcal{C}_\ell \right\} = \mathscr{C}_k.$$

In addition, a server-less CRP component $\mathbb{C}_\kappa = (\{i\}, \emptyset)$ consisting of a single service class $i$ is part of the set of service classes uniquely served by the set $U_{\mathscr{S}_k}(M)$ if and only if all the CRP components $k'$ such that $\mathbb{C}_\kappa$ has a directed arc to $\mathbb{C}_{k'}$ in the DAG $\mathcal{D} = ([K], \mathcal{A})$ are included in $(\sigma(1), \ldots, \sigma(k))$. Recalling the definition of the function $\mathsf{comps}(\sigma, \cdot)$, this is equivalent to saying that $\mathsf{comps}^{-1}(\sigma, \kappa) \leq k$.

Part (iv) follows from the definition of slack $\Delta()$ and part (iii):

$$\Delta(\mathscr{S}_k) = \mu_{\mathscr{S}_k} - \lambda_{U_{\mathscr{S}}(M)} = \sum_{\ell=1}^{k} \mu_{\mathcal{S}_\ell} - \sum_{\ell=1}^{k} \sum_{\kappa \in \mathsf{comps}(\sigma,\ell)} \lambda_{\mathcal{C}_\kappa}$$
$$= \sum_{\ell=1}^{k} \sum_{\kappa \in \mathsf{comps}(\sigma,\ell)} \mu_{\mathcal{S}_\kappa} - \lambda_{\mathcal{C}_\kappa} =: \epsilon \sum_{\ell=1}^{k} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)} + o(\epsilon).$$

□

Proof of Lemma 6: The first part follows from the proof of (Afèche et al., 2021, Lemma 4) where it is argued that if the subset $S = \{s_1, \ldots, s_\ell\}$ does not obey the condition mentioned, then

$$\mu_S - \Lambda_{U_S(M)} > 0,$$

and hence $\lim_{\epsilon \to 0} \frac{\epsilon}{\Delta(S)} = 0$. The second part follows from part (iv) of Lemma 4. □

Proof of Proposition 7: The proof of the first part of the Proposition follows exactly the same lines as (Afèche et al., 2021, Proposition 2) and hence we omit it. The calculations for the second part are as follows. Fix a topological ordering $\sigma \in \mathcal{T}(\mathcal{D}, K')$, sub-permutations $\mathbf{s}_k \in \Sigma_{\mathcal{S}_k}$, and $s = (\mathbf{s}_{\sigma(1)} || \cdots || \mathbf{s}_{\sigma(K')})$. For succinctness, define $m_k$ for $k \in \{0, 1, \ldots, K'-1\}$ by

$$m_0 = 0, \quad \text{and} \qquad m_\ell = m_{\ell-1} + |\mathcal{S}_{\sigma(\ell-1)}|.$$

From (2.26)

$$\pi(P(s;m)) = \mathcal{B} \prod_{\ell=1}^{m} \frac{1}{\Delta(s_1, \ldots, s_\ell)}$$
$$= \mathcal{B} \prod_{k=1}^{K'} \left( \prod_{\ell=m_{k-1}+1}^{m_k - 1} \frac{1}{\Delta(s_1, \ldots, s_\ell)} \right) \cdot \frac{1}{\Delta(s_1, \ldots, s_{m_k})}.$$

By Lemma 6,

$$\lim_{\epsilon\to 0}\frac{\epsilon}{\Delta(s_1,\ldots,s_{m_k})}=\frac{1}{\sum_{i=1}^{k}\widetilde{\gamma}_{\mathsf{comps}(\sigma,i)}}.$$

For some $k\in[K']$, and $m_{k-1}+1\le\ell\le m_k-1$, denote $S=\{s_{m_{k-1}+1},\ldots,s_\ell\}$. Following the outline in (Afèche et al., 2021, Lemmas 5 and 8), it follows that:

$$\lim_{\epsilon\to 0}\Delta(s_1,\ldots,s_\ell)=\mu_S-\Lambda_{U_S(\breve{M})}>0.$$

For $\mathbf{s}_k=(s_k(1),\ldots.s_k(|\mathcal{S}_k|))\in\Sigma_{\mathcal{S}_k}$, denote

$$\theta_k(\mathbf{s}_k)=\prod_{\ell=1}^{|\mathcal{S}_k|-1}\frac{1}{\mu_{\{s_k(1),\ldots,s_k(\ell)\}}-\Lambda_{U_{\{s_k(1),\ldots,s_k(\ell)\}}(\breve{M})}}. \tag{2.A1}$$

Then,

$$\begin{aligned}\lim_{\epsilon\to 0}\pi(P(s;m))&=\lim_{\epsilon\to 0}\frac{\mathcal{B}}{\epsilon^{K'}}\prod_{k=1}^{K'}\left(\prod_{\ell=m_{k-1}+1}^{m_k-1}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\cdot\frac{\epsilon}{\Delta(s_1,\ldots,s_{m_k})}\\&=\mathcal{B}'\left(\prod_{k=1}^{K'}\frac{1}{\sum_{i=1}^{k}\widetilde{\gamma}_{\mathsf{comps}(\sigma,i)}}\right)\left(\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right)\\&=\mathcal{B}'\cdot\mathbb{Q}(\sigma)\cdot\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k),\end{aligned}$$

where $\mathcal{B}'=\lim_{\epsilon\to 0}\mathcal{B}\epsilon^{-K'}$. □

Proof of Lemma 7: Let $s=(\mathbf{s}_{\sigma(1)}||\cdots||\mathbf{s}_{\sigma(K')})=(s_1,\ldots,s_m)\in\Sigma_m$ be induced by topological order $\sigma\in\mathcal{T}(\mathcal{D},K')$, and define $m_\ell$ for $\ell\in\{0,1,\ldots,K'-1\}$ by

$$m_0=0,\quad\text{and}\qquad m_\ell=m_{\ell-1}+|\mathcal{S}_{\sigma(\ell-1)}|.$$

Define $j(s,i) = \min\{\ell : i \in U(s_1,\ldots,s_\ell)\}$, and define $\kappa$ satisfying $m_{\kappa-1}+1 \le j \le m_\kappa$. Then, using Lemma 5, we have

$$\lim_{\epsilon\to 0} \epsilon \cdot W_i(s;m) = \lim_{\epsilon\to 0} \sum_{\ell=j(s,i)}^{m} \frac{\epsilon}{\Delta(s_1,\ldots,s_\ell)}$$

and since each of $\lim_{\epsilon\to 0} \frac{\epsilon}{\Delta(s_1,\ldots,s_\ell)}$ exists by Lemma 6,

$$\begin{aligned}
&= \sum_{\ell=j(s,i)}^{m} \lim_{\epsilon\to 0} \frac{\epsilon}{\Delta(s_1,\ldots,s_\ell)} \\
&= \sum_{k=\kappa}^{K'} \lim_{\epsilon\to 0} \frac{\epsilon}{\Delta(s_1,\ldots,s_{m_k})} + \sum_{\substack{j(s,i)\le\ell\le m,\\ \nexists k\,:\,\ell=m_k}} \lim_{\epsilon\to 0} \frac{\epsilon}{\Delta(s_1,\ldots,s_\ell)} \\
&= \sum_{k=\kappa}^{K'} \frac{1}{\sum_{\ell=1}^{k} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}.
\end{aligned}$$

The last equality follows because the second term in the preceding expression is 0 by Lemma 6, and each of the terms in the first sum is precisely of the form (2.27) in Lemma 6. The Lemma now follows by noting that $\kappa$ only depends on the CRP component $\mathbb{C}_k$ that service class $i$ belongs to and therefore so does the last expression, and $\kappa = \mathsf{comps}^{-1}(\sigma,k)$. □

## 2.E Section 2.6.2 Proofs

Proof of Lemma 8: Let $\mathcal{S}'$ be the set of all server permutations that are not induced by any topological order. Let $s$ be a server permutation induced by some topological order $\sigma \in \mathcal{T}(\mathcal{D},K')$.

We know from flow balance that

$$\lim_{\epsilon\to 0}\sum_{s'\in\mathcal{S}'}\sum_{b=0}^{m}\pi(P(s',b)) \geq \lim_{\epsilon\to 0}\sum_{x\in P(s,m)}\pi(x)q_{ij}^{0}(x).$$

□

But Proposition 7 tells us that

$$\lim_{\epsilon\to 0}\sum_{s'\in\mathcal{S}'}\sum_{b=0}^{m}\pi(P(s',b)) = 0.$$

Since $\pi(x) \in [0,1]$ and $q_{ij}^{0}(x) \in [0,1]$ for all $i \in [n]$, $j \in [m]$, and $x \in P(s,m)$, this means that

$$\lim_{\epsilon\to 0}\sum_{x\in P(s,m)}\pi(x)q_{ij}^{0}(x) = 0.$$

Proof of Lemma 9: Recall from Definition 9 that since the permutation of servers $s$ is induced by the topological order $\sigma$, we can express $s$ as the concatenation of sub-permutations:

$$s = \left(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_{\sigma(K')}\right)$$

with $\mathbf{s}_\kappa \in \Sigma_{\mathcal{S}_\kappa}$ denoting a permutation of the servers $\mathcal{S}_\kappa$ of CRP component $\mathbb{C}_\kappa$.

For $\mathbf{s}_\kappa = (s_\kappa(1),\ldots.s_\kappa(|\mathcal{S}_\kappa|)) \in \Sigma_{\mathcal{S}_\kappa}$, denote

$$\theta_\kappa(\mathbf{s}_\kappa) = \prod_{\ell=1}^{|\mathcal{S}_\kappa|-1}\frac{1}{\mu_{\{s_\kappa(1),\ldots,s_\kappa(\ell)\}} - \Lambda_{U_{\{s_\kappa(1),\ldots,s_\kappa(\ell)\}}(\breve{M})}}.$$

Also denote for $s_k \in \Sigma_k$

$$H_{ij}(s_k) = \lim_{\epsilon \to 0} \sum_{r=\hat{j}}^{|\mathcal{S}_k|-1} \left[ \left( \prod_{u=\hat{j}}^{r} \frac{1}{\Delta_j(s_1, \ldots, s_u)} \right) \left( \prod_{\ell=r+1}^{|\mathcal{S}_k|-1} \frac{1}{\Delta(s_1, \ldots, s_\ell)} \right) \right.$$
$$\left. \times \left( \frac{1}{\Delta(s_1, \ldots, s_r)} - \frac{1}{\Delta_j(s_1, \ldots, s_r)} \right) \right] + \prod_{u=\hat{j}}^{|\mathcal{S}_k|} \frac{1}{\Delta_j(s_1, \ldots, s_u)}$$

and

$$G_{ij}(s_k) = \lim_{\epsilon \to 0} \frac{1}{\Delta_j(s_k(1), \ldots, s_k(|\mathcal{S}_k|)} \prod_{u=\hat{j}}^{|\mathcal{S}_k|} \frac{1}{\Delta_j(s_1, \ldots, s_u)}.$$

Finally also recall the definition of $\mathbb{Q}(\sigma)$ from Equation (2.14) as

$$\mathbb{Q}(\sigma) = \prod_{\kappa \in [K']} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t, \ell)}}.$$

This lets us write $q_{ij}(P(s,m)) = \lim_{\epsilon \to 0} q_{ij}^{(\epsilon)}(P(s,m))$ as

$$q_{ij}(P(s,m)) = \frac{\mathcal{B}' \lambda_i}{\pi(P(s,m))} \mathbb{Q}(\sigma) \left( \prod_{\kappa \neq k} \theta_\kappa(s_\kappa) \right) H_{ij}(s_k)$$
$$- \lim_{\epsilon \to 0} \left[ \frac{\epsilon \mathcal{B}' \lambda_i}{\pi(P(s,m))} \left( \prod_{\kappa \neq k} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma, \ell)}} \right) \left( \prod_{\kappa \neq k} \theta_\kappa(s_\kappa) \right) G_{ij}(s_k) + o(\epsilon), \right] \tag{2.A1}$$

where $\mathcal{B}' = \lim_{\epsilon \to 0} \mathcal{B} \epsilon^{-K'}$. □

Proof of Lemma 10: From Proposition 7, we know that

$$\lim_{\epsilon \to 0} \pi(P(s,m)) = \mathcal{B}' \cdot \mathbb{Q}(\sigma) \prod_{k=1}^{K'} \theta_k(\mathbf{s}_k), \tag{2.A2}$$

where $\theta_k(s_k)$ is given by Equation (2.A1).

From the definition of $P_k(\mathbf{s}_k)$,we have that

$$\pi(P_k(\mathbf{s}_k)) = \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')} \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_k||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_\kappa}\}_{\kappa\in[K']}}} \pi(P(s,m)). \tag{2.A3}$$

This means that

$$\begin{aligned}\lim_{\epsilon\to 0}\pi_M(P(s_k)) &= \mathcal{B}'_M \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')} \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_k||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_\kappa}\}_{\kappa\in[K']}}} \mathbb{Q}(\sigma)\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k) \\ &= \mathcal{B}'_M \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\left[\mathbb{Q}(\sigma) \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_k||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_\kappa}\}_{\kappa\in[K']}}} \prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right]\end{aligned} \tag{2.A4}$$

Since the values of $\theta_\kappa(s_\kappa)$ are independent of each other and do not depend on $\sigma$, we can rewrite this as

$$\lim_{\epsilon\to 0}\pi_M(P(s_k)) = \mathcal{B}'_M\cdot\theta_k(s_k)\left(\sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\mathbb{Q}(\sigma)\right)\prod_{\kappa\neq k}\sum_{\mathbf{s}_\kappa\in\Sigma_{\mathcal{S}_\kappa}}\theta_\kappa(\mathbf{s}_\kappa) \tag{2.A5}$$

Recall from Section 2.6.1

$$\left(\mathcal{B}'_M \sum_{\{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{k\in[K']}}\prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right) = \frac{1}{\sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\mathbb{Q}(\sigma)}.$$

This lets us rewrite $\mathcal{B}'_M$ as

$$\mathcal{B}'_M = \frac{1}{\left(\prod_{\kappa=1}^{K'}\sum_{\{\mathbf{s}_\kappa\in\Sigma_{\mathcal{S}_\kappa}\}}\theta_\kappa(\mathbf{s}_\kappa)\right)\sum_{\sigma\in\mathcal{T}(\mathcal{D},K')}\mathbb{Q}(\sigma)}$$

Substituting this back into Equation (2.A5), we have that

$$\lim_{\epsilon \to 0} \pi(P_k(\mathbf{s}_k)) = \frac{\theta_k(\mathbf{s}_k)}{\sum_{\mathbf{s}_\kappa \in \Sigma_{\mathcal{S}_k}} \theta_\kappa(\mathbf{s}_\kappa)}. \tag{2.A6}$$

But $\theta_k(\mathbf{s}_k)$ depend only on $\Lambda$, $\mu$, and $\breve{M}$, for all $k \in [K']$, proving the result. □

# CHAPTER 3

# DESIGNING SERVICE MENUS FOR BIPARTITE QUEUEING SYSTEMS WITH STRATEGIC CUSTOMERS

## 3.1 Introduction

In this chapter, we extend the model in Chapter 2 to allow for customers to strategically choose which service classes to join when they arrive into the system. We consider the question of how to design a queueing system in a multi-class multi-server environment when customers are acting strategically. In particular, we will the consider problem of designing a queueing matching system such as the one depicted in Figure 3.1. In this system, customers of different types $\theta = 1, \ldots, \Theta$ arrive to the system at rates $\alpha_\theta$ seeking service by one of many available servers $j = 1, \ldots, m$. (A detailed mathematical formulation is provided in Section 3.2.) Servers are heterogeneous in terms of the amount of time it takes them to serve

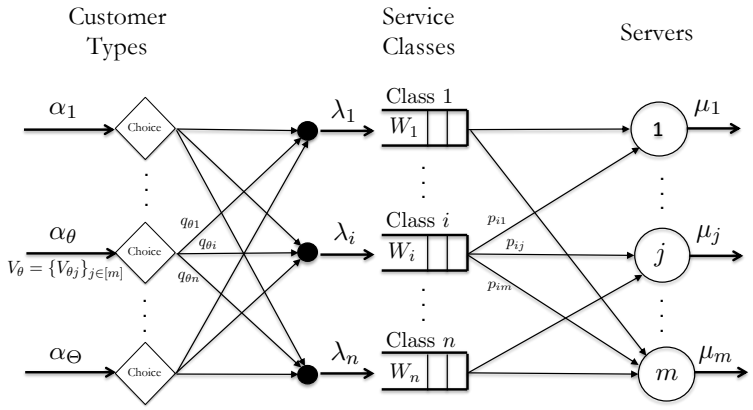



Figure 3.1: A multi-class multi-server matching queueing system.

a customer (i.e., have different service rates $\mu_j$) as well as on other attributes that affect the reward $\{V_{\theta j}\}$ that customers receive for the service.

The goal of the service provider is to design a service mechanism that will match customers to servers and will balance two (usually) competing objectives: (1) maximize customers' average matching service reward and (2) minimize customers' average waiting time. We will restrict ourselves to a special class of mechanisms in which the service provider offers a static menu of service classes $i = 1, \ldots, n$ and customers choose which one of them to join upon arrival. A service class is defined by a single queue served by a specific subset of servers under a FCFS-ALIS service discipline. Upon arrival, customers must choose which service class to join. The decision is irrevocable, so no jockeying among queues or reneging is allowed. A strategy for a type-$\theta$ customer is defined by a probability distribution $\{q_{\theta i}\}$ over the service classes, with $q_{\theta i}$ being the probability that a type-$\theta$ customer joins service class $i$. We assume that customers act as rational self-interested agents when choosing their strategies by maximizing their expected net utility, which is given by the difference between the expected server-dependent service reward they receive and a disutility waiting cost based on the mean steady-state waiting time $W_i$ of the service class they join. The expected server-dependent service reward that a customer gets from joining a service class depends on the steady-state matching probabilities $\{p_{ij}\}$ that determine the likelihood that a customer who joins class $i$ will be served by server $j$ in equilibrium under the FCFS-ALIS service discipline.

To illustrate some of the features of the problem at hand, let us consider a concrete example with two servers ($m = 2$). In this setting, the service provider can offer one of the five different service menus in Figure 3.2. For example, she can offer a *Dedicated menu*

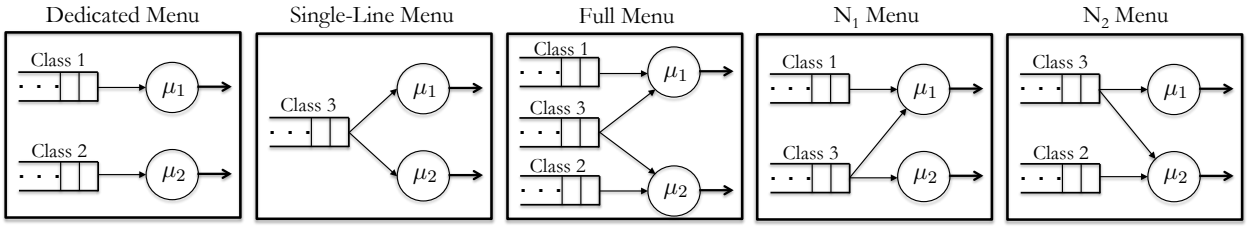



Figure 3.2: Possible service menus in a system with two servers.

(far-most left panel) consisting of two service classes (queues) each served exclusively by one of the two servers. Alternatively, the service provider can offer a *Full menu* (middle panel) in which customers have three options; they can choose between two dedicated service classes each served exclusively by one of the two servers or they can join a third class served by both servers. A customer who chooses this third class does not know with certainty which one of the two servers will be the one providing the service. The service provider can also offer a $N_i$ *menu* for $i = 1, 2$, (right two panels), in which the customers have two options; they can choose between a dedicated service class served exclusively by server $i$, or they can join a class served by both servers.

Figure 3.3 depicts an example of the equilibrium performance of the five menus in Figure 3.2 in the average reward vs. average delay quadrant for different values of the system utilization $\rho$. A complete analysis of the two-server case is presented in Section 3.5.

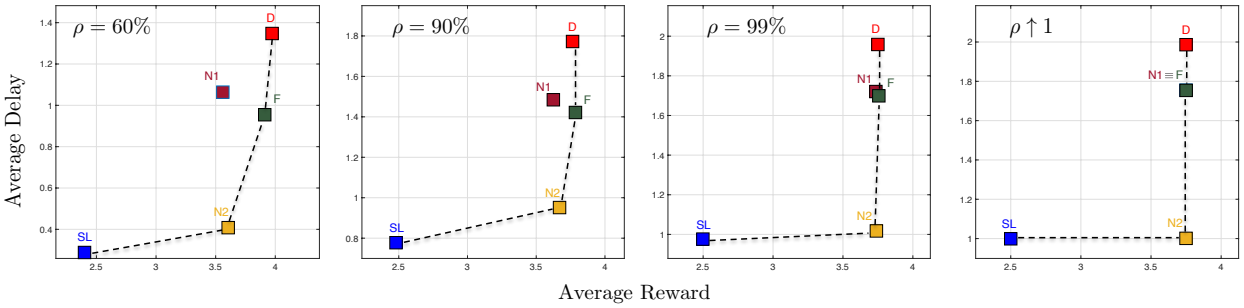



Figure 3.3: Equilibrium performance of the five menus (Dedicated (D), Single Line (SL), Full (F), $N_1$ and $N_2$) in the average reward vs. average delay quadrant for different values of the system utilization $\rho$.

For example, we will show that if the service provider only cares about minimizing customers' average waiting times, then the *Single-Line menu* is an optimal menu (see Theorem 4), which is something to be expected since a single line guarantees complete resource pooling. On the flip side, if the service provider is exclusively interested in maximizing average matching rewards and pays no attention to waiting times, then the *Dedicated menu* is an optimal menu (see Theorem 5) under heavy traffic conditions.

It is worth noticing that in this example, we have chosen the parameters of the model in such a way that menu $N_2$ dominates the other four menus when the system operates in heavy traffic $\rho \uparrow 1$ (right-most panel). Thus, in this case, it is possible to select a menu that achieves first best performance in both measures simultaneously (see Remark 4 for details). The model parameters we have chosen to achieve this are such that there is exactly one customer type that prefers server 1 to server 2, and the arrival rate of that customer type exceeds the capacity of server 1. With parameters satisfying these conditions, even the customer type who most prefers server 1 needs to be served partially by server 2 in order for the system to have stable queue lengths, and so there is no need for the service provider to offer service class served only by server 1, and thus the service provider has enough service pooling to achieve the minimum possible heavy traffic delays. By still offering a class served only by server 2, the service provider is able to divert customers who prefer server 2 to server 1 away from server 1, allowing all of server 1's capacity to be used to serve customers who prefer server 1, thus achieving the maximum possible matching value.

Interestingly, since the $N_2$ menu is a restricted version of the Full menu in which service class 1 is not offered, this example shows that reducing customers' choices can lead to more efficient outcomes (a form of Braess's paradox).

The rest of this chapter is organised as follows. In Section 3.2, we formally introduce the model and the notion of equilibrium we use. In Section 3.3, we show some of the features and challenges of this model when there are only two servers. We find that even when there are only two servers, we are unable to solve the problem analytically and instead must rely on identifying equilibria computationally. While it is possible to solve the problem computationally for two servers, the approach does not extend to the general case. This motivates the development of a strategic equilibrium in heavy-traffic that we present in Section 3.4. In Section 3.5 we return to our two server example, and complete the equilibrium analysis and consider the menu design problem in heavy-traffic. In Section 3.6, we investigate

conditions under which a first best menu exists in a system with an arbitrary number of servers. We show that in the extreme cases in which the service provider's sensitivity to delay $\zeta$ is either zero or infinity an optimal menu is given by a Single-Line or Dedicated menu, respectively. For an arbitrary value of $\zeta$, we derive necessary (Theorem 7) and sufficient (Theorem 8) conditions for a first best outcome to be achieved, which are based on the solution to a max-flow problem.

Following this, we develop three classes of menus that perform well, though not necessarily optimally. In Section 3.7, we study a special class of *Partition menus* in which the set of servers are partitioned into pools of servers, each acting as a 'super-server' that serves a single service class. One of the key advantages of partition menus is that they are very simple to explain and implement in practice. Furthermore, despite their limitations, partition menus have a number of desirable theoretical properties (e.g., they include both delay-minimizing and reward-maximizing menus) that the service provider can use to balance the trade-off between waiting times and matching values. Furthermore, they are also tractable from a computational standpoint and we exploit this in Section 3.7.3 to propose a mixed-integer linear program (MILP) that finds an optimal partition menu. In Section 3.8 we adopt a *mechanism design* approach to tackle the problem of finding optimal service menus. Specifically, we formulate MILPs that design a service class or set of service classes for each customer type to maximize a certain objective in an incentive compatible way. We call menus we design using this approach *Tailored menus*, as every service class is tailored to a specific customer type. We look at two classes of Tailored menus. In Section 3.8.1 we develop a MILP formulation that finds a delay-minimizing tailored menu among those that generate the maximum matching value. In Section 3.8.2 we take the opposite point of view and formulate a MILP that finds a value-maximize menu among a particular subset of menus that support complete resource pooling and therefore have minimum mean delay. Finally, in Section 3.9 we conduct a set of numerical experiments to compare the performance of

Partition and Tailored menus as a function of different parameters of the model including the matrix of matching rewards $V$ and the service provider's sensitivity to delay $\zeta$. Concluding remarks and discussion of future research directions can be found in Chapter 4.

## 3.2 Model Description

In this section, we provide a detailed mathematical description of the queueing service system depicted in Figure 3.1. The system is composed of three main components: (i) the stochastic model governing the queueing dynamics of the various service classes and servers' service discipline, (ii) the strategies that customers use to choose which service class to join upon arrival, and (iii) the service provider's objective criteria (and corresponding optimization problem) that are used to select an optimal menu of service classes to offer. (The service provider selects this menu at time $t = 0$ and keeps it fixed from then on.) We next discuss these three components in tandem.

**(i) Queueing Model:** A collection of $\Theta \in \mathbb{N}$ customer types arrives to the service system over time according to independent Poisson processes with rates $\alpha = (\alpha_\theta)_{\theta \in [\Theta]}$. The system is composed of $m$ servers and $n$ service classes, where each service class $i \in [n]$ is defined by a subset $S_i \subseteq [m]$ of the $m$ available servers. Server $j \in [m]$ has an exponentially distributed service time with rate $\mu_j$ and we let $\mu = (\mu_j)_{j \in [m]}$ denote the vector of service rates of the $m$ servers.

The collection of service classes constitutes a *service menu*, which can be expressed by a binary compatibility matrix $M \in \{0,1\}^{n \times m}$, where the entries of $M$ specify which service classes can be served by which servers. That is, service class $i$ can be served by server $j$ if and only if $m_{ij} = 1$ for $i \in [n]$ and $j \in [m]$. Using a slight abuse of terminology and notation, we refer to the matrix $M$ as a service menu and use $i \in M$ to denote that service class $i$ belongs to service menu $M$.

We assume that upon arrival and before observing the queue lengths, customers select one service class and join the queue of this class and wait to be served according to a FIFO queueing discipline. This decision is irreversible, that is, after joining a queue, the customer stays in it until the service is completed. Servers, on the other hand, serve the different classes using a FCFS service discipline among all compatible service classes, that is, an idle server $j \in [m]$ serves the customer that has been waiting the longest among all the service classes $i \in M$ such that $j \in S_i$. We also assume that a customer who arrives to find idle compatible servers (and hence also an empty queue) will be routed to the compatible server that has been idle the longest. Under these conditions, we say that the queueing system operates under a FCFS-ALIS (first come first served - assign longest idle server) service discipline.

**(ii) Customers' Strategies:** For a given service menu $M \in \{0,1\}^{n\times m}$, a *strategy* for an arriving customer type $\theta$ is a probability distribution $q_\theta = (q_{\theta i})_{i\in M}$ over the set of service classes in $M$, where $q_{\theta i}$ is the probability that a type-$\theta$ customer selects to join class $i \in M$. We let $q = \{q_\theta\}_{\theta\in\Theta}$ denote the strategy profile of all customer types and let $Q(M)$ be the set of all feasible strategy profiles for a given menu $M$, that is,

$$Q(M) := \Big\{ q \in \mathbb{R}_+^{\Theta\times n} \colon \sum_{i\in M} q_{\theta i} = 1 \text{ for all } \theta \in [\Theta] \Big\}.$$

A strategy profile $q \in Q(M)$ induces a vector of arrival rates $\lambda(q) = (\lambda_i(q))_{i\in M}$ to each service class, where

$$\lambda_i(q) = \sum_{\theta\in[\Theta]} \alpha_\theta \, q_{\theta i} \qquad i \in M.$$

Indirectly, through the vector of arrival rates $\lambda(q)$, the strategy profile $q \in Q(M)$ also determines the vector $W(q) = (W_i(q))_{i\in M}$ of steady-state waiting times for each service class as well as the matrix of matching probabilities $p(q) = (p_{ij}(q))$ between service classes

and servers, where $p_{ij}(q)$ denotes the steady-state probability that a customer joining class $i \in M$ will be served by server $j \in [m]$ under the strategy profile $q$. We will restrict our attention to menus $M$ and strategies $q \in Q(M)$ that are stable in the sense that they jointly admit a well-defined steady state for the service system. The condition for a menu to be stable in terms of the arrival rates into the service classes $\lambda(q)$ and the service rates of the different servers $\mu$ is given in Proposition 1.

We will further restrict attention to *admissible* menus $M$ for which the stability condition above is satisfied for some feasible strategy profile $q \in Q(M)$. Note that this is a different notion of admissibility than we used in Chapter 2. We will not be using the notion of admissibility from Chapter 2 in this chapter.

**Definition 10.** (Admissible Menus and Strategies) *A menu $M$ is* admissible *if there exists a strategy profile $q \in Q(M)$ for which the service system is stable, that is, the inequalities in Proposition 1 are satisfied. We let $\mathcal{M}$ denote the set of admissible menus.*

*For an admissible menu $M \in \mathcal{M}$, we denote by $\mathcal{Q}(M) \subseteq Q(M)$ the set of all feasible strategy profiles for which the service system is stable.*

A necessary and sufficient condition for $\mathcal{M}$ to be non-empty is that the cumulative arrival rate is strictly less than the cumulative service capacity, $|\alpha| < |\mu|$. Indeed, under this condition, it is not hard to see that any menu $M$ in which every server is connected to at least one service class is admissible. In particular, the single-line menu (i.e., $n = 1$ and $m_{1j} = 1$ for all $j \in [m]$) is admissible.

We will now look at how customers choose their strategies. We assume that customers have heterogeneous preferences over the servers and we denote by $V_\theta = (V_{\theta j})_{j \in [m]}$, the vector of rewards for a type-$\theta$ customer, where $V_{\theta j}$ is the reward that a customer $\theta$ gets when served by server $j$. We further assume that customers incur a per-unit cost of waiting $\delta$, which is homogeneous across customer types. The parameter $\delta$ captures the customers' sensitivity to

delay. We then assume the expected utility that a type-$\theta$ customer gets from joining class $i$ is then equal to

$$U_{\theta i}(W,p) := \sum_{j\in S_i} p_{ij}\, V_{\theta j} - \delta\, W_i.$$

Given a pair $(W,p)$ of steady-steady waiting times and matching probabilities, a rational utility-maximizing type-$\theta$ customer joins the service class $i$ that maximizes $U_{\theta i}(W,p)$ in equilibrium.

**Definition 11.** ($\Delta$-Equilibrium and Equilibrium Profiles) *Let $M \in \mathcal{M}$ and let $q^* \in \mathcal{Q}(M)$ be a strategy profile with corresponding vector of waiting times $W^* = W(q^*)$ and matrix of matching probabilities $p^* = p(q^*)$.*

–) $\Delta$-Equilibrium Profile*: For a given $\Delta \geq 0$, we say that $(q^*, W^*, p^*)$ is a* $\Delta$-equilibrium *profile if for all $\theta \in [\Theta]$ and for all $i,k \in [n]$*

$$q^*_{\theta i}\left(U_{\theta i}(W^*,p^*) - U_{\theta k}(W^*,p^*)\right) + \Delta \geq 0.$$

*We let $\mathcal{Q}^{\Delta}(M)$ be the set of strategies $q^*$ for which an $\Delta$-equilibrium profile $(q^*, W^*, p^*)$ exists.*

–) Equilibrium Profile*: We say that $(q^*, W^*, p^*)$ is an* equilibrium *profile if it is a 0-equilibrium profile. We let $\mathcal{Q}^*(M)$ be the set of strategies $q^*$ for which an equilibrium profile $(q^*, W^*, p^*)$ exists.*

Trivially, every equilibrium profile is a $\Delta$-equilibrium profile for all $\Delta > 0$. We also note that in an equilibrium profile $(q^*, W^*, p^*)$ if $q^*_{\theta i} > 0$ for some customer type $\theta \in \Theta$ and service class $i \in [n]$ then we must have that $U_{\theta i}(W^*,p^*) \geq U_{\theta k}(W^*,p^*)$ for all service classes $k \in [n]$. That is, the expected utility that customer type $\theta$ gets from joining class $i$ is at least as large as the expected utility that the customer would get from joining any other service class $k$. In other words, in a equilibrium we can have $q^*_{\theta i} > 0$ only if $U_{\theta i}(W^*,p^*) =$

$\max_{k\in[n]}\{U_{\theta k}(W^*, p^*)\}$.

The following theorem guarantees the existence of equilibrium profiles when the system has sufficient service capacity to serve all of the customers.

**Theorem 3.** *Suppose that $|\alpha| < |\mu|$, and $M \in \mathcal{M}$ is an admissible service menu. Then there exists an equilibrium strategy profile $q^* \in \mathcal{Q}^*(M)$.*

The proof of the theorem can be found in Section 3.A.1, and is based on a fixed-point argument.

**(iii) Optimal Service Menu:** The final component of the model corresponds to the objective that the service provider uses to select an optimal menu $M^* \in \mathcal{M}$. Similar to the preferences of individual customers, we assume that the service provider is interested in maximizing the value generated by the matching between customers and servers while minimizing the waiting time experienced by these customers. Specifically, for a given admissible menu $M$ and customers' strategy $q \in \mathcal{Q}(M)$, we assume that the service provider collects a payoff equal to

$$\Pi(M, q) := \overline{V}(M, q) - \zeta\, \overline{W}(M, q), \tag{3.2.1}$$

where $\zeta$ is a positive scalar that captures the service provider's sensitivity to customers' delays and

$$\overline{V}(M, q) := \sum_{\theta\in[\Theta]} \sum_{i\in[n]} \sum_{j\in[m]} \alpha_\theta\, q_{\theta i}\, p_{ij}(q) V_{\theta j} \qquad \text{and} \qquad \overline{W}(M, q) := \sum_{i\in[n]} \lambda_i(q)\, W_i(q)$$

correspond to the cumulative steady state matching reward and waiting time, respectively, experienced by all customers.

It is worth noticing that while the service provider selects the service menu $M$, it is the customers who decide which service classes they want to join by selecting an equilibrium

strategy $q^* \in \mathcal{Q}^*(M)$. Hence, from a game theoretic standpoint, the service provider acts as Stackelberg leader who moves first by selecting the service menu $M \in \mathcal{M}$ to offer at time 0 and then the arriving customers respond to the specific choice of $M$ by selecting a best-response strategy $q^* \in \mathcal{Q}^*(M)$ that maximizes their expected utility according to Definition 11. Hence, the service provider's optimization problem can be formulated as follows:

$$\sup_{M \in \mathcal{M}} \sup_{q^* \in \mathcal{Q}^*(M)} \Pi(M, q^*). \tag{3.2.2}$$

**Remark 1.** Formulation (3.2.2) assumes that the service provider is able to select which equilibrium strategy $q^* \in \mathcal{Q}^*(M)$ customers' will end up playing. This is, of course, without loss of generality for those admissible menus $M$ for which $\mathcal{Q}^*(M)$ is a singleton. However, when $M$ induces multiple equilibria formulation (3.2.2) models the problem of an 'optimistic' service provider. Alternatively, we could have adopted a pessimistic view by formulating the service provider's problem as follows:

$$\sup_{M \in \mathcal{M}} \inf_{q^* \in \mathcal{Q}^*(M)} \Pi(M, q^*). \quad \diamond$$

**Remark 2.** (Social Planner) If $\zeta = \delta$ then $\Pi(M, q) = \sum_{\theta \in [\Theta]} \alpha_\theta \, \overline{U}_\theta(q)$, where $\overline{U}_\theta(q)$ is the expected utility of a $\theta$ customer under strategy $q$, that is, $\overline{U}_\theta(q) := \sum_{i \in [n]} q_{\theta i} U_{\theta i}(W(q), p(q))$. In other words, when $\zeta = \delta$ the service provider acts as a social planner who is interested in maximizing the cumulative utility of all customers. $\diamond$

## 3.3 Service Menus with Two Servers

In this section we illustrate the model and solution to the service provider's problem in (3.2.1) by characterizing optimal service menus for the special case in which the system has two servers (i.e., $m = 2$). In this setting, we are able to obtain a complete solution as a

function of the model's parameters, which provides a number of insights that we will use later to analyze the general case with an arbitrary number of severs. The two-server model is also worth studying in its own right as it provides a parsimonious framework that allows for a non-trivial segmentation of service (e.g., high vs. low quality or fast vs. slow service).

With two servers, there are three possible service classes, namely, Class 1 served only by server 1, Class 2 served only by server 2, and Class 3 served by both servers. With these three classes available, the service provider can offer one of the following five admissible service menus (see Figure 3.2):[1]

- DEDICATED MENU (D), in which Classes 1 and 2 are offered,
- SINGLE-LINE MENU (SL), in which only service Class 3 is offered,
- FULL MENU (F), in which all three classes are offered,
- $N_i$ MENU, in which Classes $i$ and 3 are both offered, for $i = 1, 2$.

**3.3.1 Performance Analysis in Steady State** In order to derive the equilibrium strategies of these menus we first need to characterize their steady-state performance in terms of waiting times and matching probabilities. To this end, let us fix the service menu $M$. Since the steady-state analysis of the Dedicated and Single Line menus reduce to those of two M/M/1 and one M/M/2 systems, respectively, we will only discuss the cases in which $M \in \{\text{F}, N_1, N_2\}$.

We derive the steady-state performance of an arbitrary strategy profile $q \in \mathcal{Q}(M)$ using the Markov chain representation of the system proposed by Adan and Weiss (2014) and its corresponding stationary distribution. The following result summarizes this derivation,

1. We note that it is possible to offer two additional menus each consisting exclusively of service Class $i$ with $i = 1, 2$. However, the menu that offers only Class $i$ is dominated by menu $N_i$.

whose statement make use of the following notation $\Lambda := |\lambda|$, $\Gamma := |\mu|$, $\Delta_i := \mu_i - \lambda_i$, for $i = 1, 2$, $\Delta := \Gamma - \Lambda$ and

$$\mathcal{B} := \left[ \frac{\Delta + \lambda_3}{\Delta\, \Delta_1\, \Delta_2} + \frac{1}{\Delta_1\,(\Lambda - \lambda_1)} + \frac{1}{\Delta_2\,(\Lambda - \lambda_2)} + \frac{\Lambda + \lambda_3}{\Lambda\,(\Lambda - \lambda_1)\,(\Lambda - \lambda_2)} \right]^{-1}.$$

**Proposition 8.** (Steady-State Performance) *Suppose $M \in \{\mathrm{F}, N_1, N_2\}$. Let $q \in \mathcal{Q}(M)$ be a fixed customers' strategy profile, which induces a vector of arrival rates $\{\lambda_i\}_{i\in[n]}$ to the service classes. Then, the steady-state probability that a customer joining Class 3 is served by server 1 and server 2 are equal to*

$$p_{31} = \mathcal{B}\left[ \frac{1}{\Lambda\,(\Lambda - \lambda_2)} + \frac{1}{\Delta_2\,(\Lambda - \lambda_2)} + \frac{1}{\Delta\,(\Gamma - \lambda_2)} \left(1 + \frac{\mu_1}{\Delta_2}\right) \right] \quad \textit{and} \quad p_{32} = 1 - p_{31}, \tag{3.3.1}$$

*respectively. The steady-state waiting times for the three services classes are given by*

$$W_1 = W_3 + \frac{\mathcal{B}}{\Delta_1^2}\left[ \frac{1}{\Lambda - \lambda_1} + \frac{1}{\Delta} \right], \quad W_2 = W_3 + \frac{\mathcal{B}}{\Delta_2^2}\left[ \frac{1}{\Lambda - \lambda_2} + \frac{1}{\Delta} \right] \quad \textit{and} \quad W_3 = \frac{\mathcal{B}\,(\Delta + \lambda_3)}{\Delta^2\,\Delta_1\,\Delta_2}. \tag{3.3.2}$$

Proof of Proposition 8: Let $X$ denote set of states of the Markov chain proposed by Adan and Weiss (2014) with $x \in X$ a generic state of this Markov chain and $\pi(x)$ its steady state probability distribution. The set $X$ is partitioned into the following subsets:

(a) $x = (s_i, n_i, s_j, n_{j3})$: Both servers are busy with server $i$ serving the oldest arrival, with $i = 1, 2$ and $j = 3 - i$. There are $n_i \geq 0$ customers waiting in the queue of Class $i$ and $n_{j3} \geq 0$ customers waiting in the queues of Classes $j$ and 3 combined. The steady-state probability of $x$ is given by

$$\pi(x) = \mathcal{B}\,\frac{\lambda_i^{n_i}(\lambda_1 + \lambda_2 + \lambda_3)^{n_{j3}}}{\mu_i^{n_i+1}(\mu_1 + \mu_2)^{n_{j3}+1}},$$

for some appropriate normalizing constant $\mathcal{B}$.

(b) $x = (s_i, n_i, s_j)$: Server $i$ is busy and server $j$ is idle. There are $n_i \geq 0$ customers waiting in the queue of Class $i$ and the queues of Classes 1 and 3 are necessarily empty. In this case,

$$\pi(x) = \mathcal{B} \frac{\lambda_i^{n_i}}{\mu_i^{n_i+1}(\lambda_j + \lambda_3)}.$$

(c) $x = (s_i, s_j)$: Both servers are idle with server $i$ being idle the longest. In this case,

$$\pi(x) = \frac{\mathcal{B}}{(\lambda_1 + \lambda_2 + \lambda_3)(\lambda_i + \lambda_3)}.$$

The value of $\mathcal{B}$ is obtained by imposing

$$\sum_{x \in X} \pi(x) = 1.$$

To alleviate the notation, let us define $\Lambda := \lambda_1 + \lambda_2 + \lambda_3$, $\Gamma := \mu_1 + \mu_2$, $\Delta_1 := \mu_1 - \lambda_1$, $\Delta_2 := \mu_2 - \lambda_2$ and $\Delta := \Gamma - \Lambda$. It follows that

$$\mathcal{B} = \left[ \frac{\Delta + \lambda_3}{\Delta\, \Delta_1\, \Delta_2} + \frac{1}{\Delta_1\,(\Lambda - \lambda_1)} + \frac{1}{\Delta_2\,(\Lambda - \lambda_2)} + \frac{\Lambda + \lambda_3}{\Lambda\,(\Lambda - \lambda_1)\,(\Lambda - \lambda_2)} \right]^{-1}. \tag{3.3.3}$$

To calculate the matching probabilities, we first calculate the rate of transitions in the Markov chain associated with a customer from service class 3 beginning service with each server. As the problem is symmetric in the servers, we will only go through the calculations to identify the rate of transitions associated with a class 3 customer beginning service with server 1, which we shall label $f_{31}$.

The FCFS-ALIS service discipline lets us immediately conclude that there are no transitions from states $(s_2, s_1)$ or $(s_1, n_1, s_2)$ that involve a class 3 customer beginning service with server 1. Any arriving class 3 customer will immediately begin service with the server

who has been idle longest, which is server 2 in both cases. In the $(s_1, n_1, s_2)$, we can also see that server 1 completing service will not trigger a class 3 customer beginning service with server 1, as the only waiting customers for server 1 to serve are those that are incompatible with server 2 (i.e., class 1 customers). Similar reasoning tells us that the transitions from state $(s_1, s_2)$ and $(s_2, n_2, s_1)$ associated with a class 3 customer beginning service with server 1 are all of those transitions resulting from a class 3 customer arriving, and hence $f_{31}$ includes the terms $\lambda_3\pi(s_1, s_2)$ and $\lambda_3\pi(s_2, n_2, s_1)$. For $n_1 > 0$, there are no transitions from $(s_1, n_1, s_2, n_2)$ that result in a class 3 customer beginning service with server 1, since as soon as a server 1 finishes serving the customer they are currently serving, they will begin serving another waiting class 1 customer. However, for $n_1 = 0$ and $n_2 > 0$, when server 1 finished serving their current customer, they will begin service with a class 3 customer if a class 3 has been waiting the longest out of those compatible with server 1. This will happen if $n_2$ consists of $x$ class 2 customers, followed by a class 3 customer. Thus $f_{31}$ will include the term $\mu_1 \sum_{n_2=1}^{\infty} \sum_{x=0}^{n_2-1} \frac{\lambda_2^x \lambda_3}{(\lambda_1+\lambda_2+\lambda_3)^{x+1}} \pi(s_1, 0, s_2, n_2)$. We can use similar reasoning to include that the transition rate also includes the term $\mu_1 \sum_{n_1=1}^{\infty} \sum_{n_2=0}^{\infty} \sum_{x=0}^{n_1-1} \frac{\lambda_1^x \lambda_3}{(\lambda_1+\lambda_2+\lambda_3)^{x+1}} \pi(s_2, n_2, s_1, n_1)$.

Thus the total rate of transitions involving a class 3 customer beginning service with server 1 is

$$f_{31} = \mathcal{B}\lambda_3 \left[ \frac{1}{\Lambda\left(\Lambda - \lambda_2\right)} + \frac{1}{\Delta_2\left(\Lambda - \lambda_2\right)} + \frac{1}{\Delta\left(\Gamma - \lambda_2\right)} \left(1 + \frac{\mu_1}{\Delta_2}\right) \right]. \tag{3.3.4}$$

The probability that a class 3 customer is server by server 1 is $p_{31} = f_{31}/\lambda_3$.

To conclude the proof, we note that the expected waiting times for the different service class can be calculated using Little's Law. □

**3.3.2 Equilibrium Strategies** The key feature of the two-server model that we exploit to derive customers' equilibrium strategies is the fact that we can rank the cus-

tomer types based on their relative preferences over the two servers. To this end, define $\Delta V_\theta := V_{\theta 2} - V_{\theta 1}$ for each customer type $\theta \in [\Theta]$ and label the elements in $[\Theta]$ by $\theta_1, \theta_2, \ldots, \theta_\Theta$ such that $\Delta V_{\theta_i} \leq \Delta V_{\theta_j}$ for all $1 \leq i < j \leq \Theta$. In case of a tie, the class that values server 2 more gets assigned a higher index.

Under this indexing, it is not hard to see that we can restrict ourselves to cut-off (threshold-type) equilibria. For example, if the service provider offers a Dedicated menu then a type $\theta$ customer (weakly) prefers Class 1 over Class 2 if $\Delta V_\theta \leq \delta\,(W_2 - W_1)$. Thus, there exists a customer type $\theta_\tau$ with $\tau \in [\Theta]$ such that all customer types $\theta_k$ with $k \leq \tau - 1$ select Class 1, all customer types $\theta_k$ with $k \geq \tau + 1$ select Class 2 and customers of type $\theta_\tau$ are indifferent and randomize between the two service classes. Similarly, if the service provider offers the Full menu then a type $\theta$ customer weakly prefers Class 1 to Class 3 if $p_{32}\,\Delta V_\theta \leq \delta\,(W_3 - W_1)$ and weakly prefers Class 2 to Class 3 if $p_{31}\,\Delta V_\theta \geq \delta\,(W_2 - W_3)$. In this case, an equilibrium involves two thresholds, $\tau_1, \tau_2 \in [\Theta]$ with $\tau_1 \leq \tau_2$. All customer types $\theta_k$ with $k \leq \tau_1 - 1$ select Class 1, all customer types $\theta_k$ with $k \geq \tau_2 + 1$ select Class 2, all customer types $\theta_k$ with $\tau_1 + 1 \leq k \leq \tau_2 - 1$ select Class 3, customers type $\theta_{\tau_i}$ are indifferent between Class $i$ and Class 3 for $i = 1, 2$.

Proposition 9 below exploits this threshold structure to characterize equilibrium strategies for the $D$, $N_1$ and $F$ menus[2]. The statement of this proposition make use of some additional notation. For $0 \leq x_1 \leq x_2 \leq \Theta$, we define

$$\lambda_1(x) := \sum_{k=1}^{\lfloor x \rfloor} \alpha_{\theta_k} + (x - \lfloor x \rfloor)\,\alpha_{\theta_{\lceil x \rceil}},$$

$\lambda_2(x_2) = |\alpha| - \lambda_1(x_2)$ and $\lambda_3(x_1, x_2) = |\alpha| - \lambda_1(x_1) - \lambda_2(x_2)$. These are the arrival rates to service classes 1, 2 and 3, respectively, if all customers type $\{1, 2, \ldots, \lfloor x_1 \rfloor\}$ plus a fraction

2. The equilibrium strategy for the Single Line is trivial and for the $N_2$ menu it can be derived from the one for the $N_1$ menu by interchanging the labels of the two servers.

$(x_1 - \lfloor x_1 \rfloor)$ of customers type $\lceil x_1 \rceil$ join Class 1, all customers type $\{\lceil x_2 \rceil + 1, \ldots, \Theta\}$ plus a fraction $(\lceil x_2 \rceil - x_2)$ of customers type $\lceil x_2 \rceil$ join Class 2, and all remaining customers join Class 3. To ensure stability, we will need to bound the values of $x_1$ and $x_2$ such that $0 \leq x_1 \leq \bar{x}_1$ and $\underline{x}_2 \leq x_2 \leq \Theta$ with

$$\bar{x}_1 := \max\Big\{0 \leq x \leq \Theta \colon \lambda_1(x) \leq \mu_1\Big\} \quad \text{and} \quad \underline{x}_2 := \min\Big\{0 \leq x \leq \Theta \colon \lambda_2(x) \leq \mu_2\Big\}.$$

Note that under the global stability condition $|\alpha| < |\mu|$ we must have $\underline{x}_2 < \bar{x}_1$. For a pair $(x_1, x_2) \in [0, \bar{x}_1) \times (\underline{x}_2, |\Theta|] \cap \{x_1 \leq x_2\}$, we define the steady-state matching probabilities $p_{3j}(x_1, x_2)$ and waiting times $W_i(x_1, x_2)$ for $i = 1, 2, 3$ and $j = 1, 2$ by replacing the values $\lambda_1(x_1)$, $\lambda_2(x_2)$ and $\lambda_3(x_1, x_2)$ in equations (3.3.1) and (3.3.2), respectively.

**Proposition 9.** *Suppose the service provider offers menus $M \in \{D, N_1, F\}$. There exists two thresholds $0 \leq x_1^* \leq x_2^* \leq \vartheta$ such that an equilibrium profile $(q^*_{\theta_k 1}, q^*_{\theta_k 2}, q^*_{\theta_k 3})$ for a type-$\theta_k$ customer satisfies*

$$q^*_{\theta_k 1} = \begin{cases} 1 & \text{if } k \leq \lceil x_1^* \rceil - 1 \\ x_1^* - \lfloor x_1^* \rfloor & \text{if } k = \lceil x_1^* \rceil \\ 0 & \text{if } k \geq \lceil x_1^* \rceil + 1 \end{cases} \qquad q^*_{\theta_k 2} = \begin{cases} 0 & \text{if } k \leq \lceil x_2^* \rceil - 1 \\ \lceil x_2^* \rceil - x_2^* & \text{if } k = \lceil x_2^* \rceil \\ 1 & \text{if } k \geq \lceil x_2^* \rceil + 1 \end{cases}$$

*and $q^*_{\theta_k 3} = 1 - q^*_{\theta_k 1} - q^*_{\theta_k 2}$. The values of $x_1^*$ and $x_2^*$ depends on the specific menu $M$ as follows:*

–) Dedicated Menu*: Let $x^* = \sup\big\{x \in (\underline{x}_2, \bar{x}_1) \colon \Delta V_{\theta_{\lceil x \rceil}} \leq \delta\big(W_2(x,x) - W_1(x,x)\big)\big\}$. If $x^* \notin \mathbb{N}$ then $x_1^* = x_2^* = x^*$. Otherwise, $x_1^* = x^* + 1$ and $x_2^* = x^*$.*

–) $N_1$ Menu*: $x_1^* = \sup\Big\{x \in [0, \underline{x}_2 \wedge \bar{x}_1) \colon\ p_{32}(x, x_2^*)\, \Delta V_{\theta_{\lceil x \rceil}} \leq \delta\big(W_3(x, x_2^*) - W_1(x, x_2^*)\big)\Big\}$ and $x_2^* = \Theta$.*

–) Full Menu*: The values of* $x_1^*$ *and* $x_2^*$ *solves the system of equations*

$$\begin{aligned} x_1^* &= \sup\left\{x \in [0, x_2^* \wedge \bar{x}_1): \quad p_{32}(x, x_2^*)\,\Delta V_{\theta_{\lceil x \rceil}} \leq \delta\left(W_3(x, x_2^*) - W_1(x, x_2^*)\right)\right\} \\ x_2^* &= \sup\left\{x \in (\underline{x}_2 \vee x_1^*, |\Theta|]: \quad p_{31}(x_1^*, x)\,\Delta V_{\theta_{\lceil x \rceil}} \leq \delta\left(W_2(x_1^*, x) - W_3(x_1^*, x)\right)\right\}. \end{aligned}$$

Proof of Proposition 9: The proof of the proposition follows from noticing that in the equilibrium of each of the three menus, some customer type(s) needs to randomise between two service classes to ensure that the equilibrium conditions are satisfied. It is easy to see that the values of $x_i^*$, $i = 1, 2$ specify precisely the customer types that need to randomise. □

An example of the equilibrium outcomes derived in Proposition 9 is depicted in Figure 3.3.

While Proposition 9 provides a complete characterization of customers' equilibrium strategies for the Dedicated, $N_1$ ($N_2$), and Full menus we can only derive these equilibria computationally for any particular set of parameters. Furthermore, as we try to move to more complex systems with an arbitrary number of servers, we are no longer able to rank customer types based on their preferences over just two servers and use the simple cut-off analysis that we have used above to derive their equilibrium strategies. For this reason, and to say something more concrete about equilibrium outcomes for general systems, we will investigate their performance under heavy traffic conditions.

## 3.4 Heavy Traffic Regime

In this section, we present the model that we will use to formally study the question of menu design through heavy-traffic asymptotics. First, in Section 3.4.1, we present the specific heavy traffic scaling of the system primitives. In Section 3.4.2 we present a quadratic programming (QP) formulation that we will use to approximate the matching probabilities

under the FCFS-ALIS service requirement. Finally, in Section 3.4.3, we introduce the notion of a *heavy traffic* equilibria, which we use to extend Definition 11 to our heavy traffic regime.

**3.4.1 Scaling** We construct a sequence of matching queueing systems parameterized by $\epsilon$ and use the superscript $(\epsilon)$ to emphasize the dependence of various quantities on $\epsilon$. For example, ${\alpha_\theta}^{(\epsilon)}$ and ${q_\theta}^{(\epsilon)} = ({q_{\theta 1}}^{(\epsilon)}, \ldots, {q_{\theta n}}^{(\epsilon)})$ are the arrival rate and strategy profile of type-$\theta$ customers in system $\epsilon$.

We assume that the bipartite matching system approaches heavy traffic as $\epsilon \downarrow 0$. Specifically, we assume that there are two vectors $A, a \in \mathbb{R}_+^{\Theta}$ (independent of $\epsilon$) with $|A| = |\mu|$ so that the sequence of arrival rates $\alpha^{(\epsilon)} = \{{\alpha_\theta}^{(\epsilon)}\}_{\theta \in \Theta}$ satisfies (for $\epsilon$ small enough):

$$ {\alpha_\theta}^{(\epsilon)} = A_\theta - a_\theta \epsilon \geq 0 \quad \text{for all } \theta \in [\Theta]. \tag{3.4.1} $$

Intuitively, in the heavy-traffic regime, the arrival rates $\alpha^{(\epsilon)}$ approach the limiting rates $A$ along the direction specified by $a$. It follows that the traffic intensity of the $\epsilon^{\text{th}}$ system equals

$$ \rho^{(\epsilon)} := \frac{\sum_\theta {\alpha_\theta}^{(\epsilon)}}{\mu_1 + \cdots + \mu_m} = \frac{|A| - |a|\,\epsilon}{|\mu|} = 1 - \frac{|a|}{|\mu|}\,\epsilon $$

and approaches one (i.e., 100% system utilization) as $\epsilon \downarrow 0$.

We let $\mathcal{M}^{(\epsilon)}$ denote the class of menus $M$ that are admissible in the sense of Definition 10. It is not hard to see that the sets $\mathcal{M}^{(\epsilon)}$ are monotonic in $\epsilon$ and so the limit $\widehat{\mathcal{M}} := \lim_{\epsilon \downarrow 0} \mathcal{M}^{(\epsilon)}$ exists. We will refer to $\widehat{\mathcal{M}}$ as the set of *admissible menus in heavy traffic*.

Under the heavy traffic condition in (3.4.1), the waiting time ${W_i}^{(\epsilon)}(q^{(\epsilon)})$ will grow out of bound as $\epsilon \downarrow 0$. For this reason, we assume that $\delta^{(\epsilon)}$ goes to 0 as $\epsilon \downarrow 0$ in such a way that the product $\delta^{(\epsilon)}\, {W_i}^{(\epsilon)}(q^{(\epsilon)})$ converges to a well-defined non-trivial limit. In particular, we will assume that $\delta^{(\epsilon)} = \delta\,\epsilon$ for some fixed constant $\delta > 0$ independent of $\epsilon$[3]. Given this scaling,

3. Alternatively, we could consider a slightly more general scaling of $\delta^{(\epsilon)}$ that only requires $\lim_{\epsilon \downarrow 0} \frac{\delta^{(\epsilon)}}{\epsilon} = \delta$.

we find convenient to define the scaled mean waiting time

$$\widehat{W_i}^{(\epsilon)}(q^{(\epsilon)}) := \epsilon \cdot W_i{}^{(\epsilon)}(q^{(\epsilon)}), \tag{3.4.2}$$

which remains bounded in heavy traffic. Finally, the expected utility of a customer type $\theta$ under strategy $q_\theta^{(\epsilon)}$ is given by

$$U_{\theta i}^{(\epsilon)}(q^{(\epsilon)}) = \sum_{j \in S_i} p_{ij}^{(\epsilon)}(q^{(\epsilon)})\, V_{\theta j} - \delta\, \widehat{W_i}^{(\epsilon)}(q^{(\epsilon)}).$$

Note that the valuations $V = [V_{\theta j}]$ and service rates $\mu = (\mu_j)$ remain constant independent of $\epsilon$.

Here we can see the benefits of assuming that the delay sensitivity $\delta^{(\epsilon)}$ goes to zero with $\epsilon$. If we did not make this assumption, then the unbounded delay costs would dominate the finite service valuation in customers' utility function, and our model would predict that customers would ignore service values and act only to minimise delays. By modelling delay sensitivity in the way we do here, the value customers receive from service and their expected delay costs remain of the same order. This means we are able to more accurately model real world scenarios with finite delays, in which customers are still accounting for service values when making choices. For the same reasons, a similar scaling is introduced for the service provider's delay sensitivity: $\zeta^{(\epsilon)} = \zeta \cdot \epsilon$ for some fixed $\zeta > 0$.

With abuse of notation, we denote the *limiting average reward* and *limiting average scaled delay* by:

$$\overline{V}^* := \frac{1}{|A|} \lim_{\epsilon \downarrow 0} \sum_{\theta \in [\Theta]} \sum_{i \in [n]} \sum_{j \in [m]} A_\theta\, q_{\theta i}^{(\epsilon)}\, p_{ij}^{(\epsilon)}(q^{(\epsilon)}) V_{\theta j} \text{ and } \overline{W}^* := \frac{1}{|A|} \lim_{\epsilon \downarrow 0} \sum_{i \in [n]} \lambda_i^{(\epsilon)}(q^{(\epsilon)})\, \widehat{W}_i^{(\epsilon)}(q^{(\epsilon)}).$$

Thus the service provider's problem in heavy-traffic is to maximise

$$\Pi(M, q^{(\epsilon)}) = \overline{V}^* - \zeta \overline{W}^*. \tag{3.4.3}$$

**3.4.2 Matching Probabilities under Heavy Traffic Equilibrium:** While the problem of computing waiting times under a FCFS-ALIS service discipline simplifies significantly under heavy traffic conditions, computing the steady-state matching probabilities $p_{ij}$ remains computational challenging due to the underlying combinatorial structure of the state-space of the system (see Adan and Weiss, 2014). In Chapter 2, we showed that in the heavy traffic limit the value of $p_{ij}$, for some service class $i$ and server $j$ that belong to the same CRP component, depends exclusively on the structure of the matching within the CRP components.

While this result reduces the problem of computing the steady-state matching probabilities under heavy traffic conditions to each individual CRP component, the combinatorial structure of the possible states within each CRP is still an obstacle for efficiently computing $p_{ij}$. Caldentey et al. (2009) and Afèche et al. (2021) identify special classes of topologies under which the matching probabilities can be computed efficiently using a quadratic program. Specifically, for a given menu $M = [m_{ij}]$ in this class of topologies, the matching probabilities $p_{ij}$ can be computed solving the following quadratic program:

$$\begin{aligned} \min_{p} \quad & \sum_{i\in[n]} \sum_{j\in[m]} \frac{\lambda_i}{\mu_j} m_{ij} p_{ij}^2 && \textbf{(QP)} \\ \text{subject to} \quad & \sum_{i\in[n]} \lambda_i m_{ij} p_{ij} = \mu_j \quad \forall j \in [m], \\ & \sum_{j\in S_i} m_{ij} p_{ij} = 1 \quad \forall i \in [m], \\ & p_{ij} \geq 0 \quad \forall (i,j) \in [n] \times [m]. \end{aligned}$$

The classes of matching topologies $M = [m_{ij}]$ for which it is known that the (**QP**) formulation produces the exact matching probabilities under a FCFS-ALIS service discipline includes:

- **Spanning Forest**: $M$ induces a spanning forest on the bipartite graph between service classes and servers;
- **Complete Graph**: $m_{ij} = 1$ for all $i \in [n]$ and $j \in [m]$;
- **Almost Complete**: $n = m$, $m_{1j} = 1$ and $m_{ij} = 1$ for $i = 2, \ldots, n$ for all $j \in [m]$;
- **Quasi-Complete**: $n = m$ and $m_{ij} = 1$ if and only if $i \neq j$.

In general, however, the (**QP**) formulation only provides an approximation to the actual FCFS-ALIS matching probabilities (see Afèche et al., 2021 and Fazel-Zarandi and Kaplan, 2018 for detailed numerical experiments) and, to the best of our knowledge, it is still unknown whether there exists a computationally efficient method to determine the exact matching probabilities for an arbitrary matching topology.

In what follows, we will proceed with our analysis using the QP formulation to compute the matching flows. The following facts about an optimal solution to (**QP**) are proven in Afèche et al. (2021).

**Proposition 10.** *Let $M$ be an admissible menu in heavy traffic, i.e., $M \in \widehat{\mathcal{M}}$. Then,*

1. *The quadratic program in* (**QP**) *is feasible and admits a unique optimal solution $p^*(M)$.*
2. *A feasible matrix of matching probabilities $p(M) = [p_{ij}(M)]$ is the optimal solution to* (**QP**) *if and only if there exist multipliers $(\omega_j,\ j \in [m])$ for the first set of constraints and $(\eta_i,\ i \in [n])$ for the second set of constraints satisfying the KKT first order stationarity conditions:*

$$p^*_{ij}(M) = \max\{\mu_j \left(\eta_i + \omega_j\right), 0\}, \quad \forall (i,j) : m_{ij} = 1.$$

The second property is particularly useful because it allows for a simple encoding of the constraints imposed by the QP formulation[4] on the matching probabilities.

**3.4.3 Heavy-Traffic Equilibrium** For a given admissible menu in heavy traffic $M \in \widehat{\mathcal{M}}$, we are interested in identifying a limiting equilibrium, as $\epsilon \downarrow 0$. To this end, we introduce the notion of a *heavy-traffic* equilibrium.

**Definition 12.** (Heavy Traffic Equilibrium) *For a given admissible menu in heavy traffic $M \in \widehat{\mathcal{M}}$, we say that $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$ is a* heavy traffic equilibrium *if there exists a vector $\hat{\phi}^* \in \mathbb{R}^{|\Theta| \times n}$ such that $\hat{q}^* + \epsilon\,\hat{\phi}^* \in \mathcal{Q}$ for all $\epsilon \geq 0$ and the following two conditions are satisfied:*

(a) Heavy Traffic Limit*: $\widehat{W}^* = \lim_{\epsilon \downarrow 0} \widehat{W}^{(\epsilon)}(\hat{q}^* + \epsilon\,\hat{\phi}^*)$ and $\hat{p}^* = \lim_{\epsilon \downarrow 0} p^{(\epsilon)}(\hat{q}^* + \epsilon\,\hat{\phi}^*)$.*

(b) Best-Response*: For all $\theta \in \Theta$ and for all $i, k \in [n]$*

$$\hat{q}^*_{\theta i} \left( U_{\theta i}(\widehat{W}^*, \hat{p}^*) - U_{\theta k}(\widehat{W}^*, \hat{p}^*) \right) \geq 0.$$

*We let $\widehat{\mathcal{Q}}^*(M)$ be the set of all strategy profiles $\hat{q}^*$ for which there exists a heavy traffic equilibrium $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$.*

Notice that when the strategies are of the form in Definition 12, strategy profiles induce a vector $\lambda^{(\epsilon)}$ of pre-limit arrival rates into service classes given by

$$\lambda_i{}^{(\epsilon)} = \sum_{\theta \in \Theta} \alpha_\theta{}^{(\epsilon)} q_{\theta i}{}^{(\epsilon)} = \sum_{\theta \in \Theta} A_\theta \hat{q}_{\theta i} - \epsilon \sum_{\theta \in \Theta} (a_\theta \hat{q}_{\theta i} - A_\theta \hat{\phi}_{\theta i}) + o(\epsilon) =: \Lambda_i - \epsilon \gamma_i + o(\epsilon), \quad (3.4.4)$$

which is exactly the scaling of pre-limit arrival rates that we need to take advantage of our performance analysis in Chapter 2.

4. Which is an approximation for the FCFS-ALIS requirements that we need to impose on $p = [p_{ij}]$.

For the notion of a heavy-traffic equilibrium to be of any practical interest, we would like to be able to map it back to some concrete equilibrium in the pre-limit. The following proposition formalizes this requirement by showing that we can always view a heavy-traffic equilibrium as the limit of a sequence of $\epsilon$-equilibria in the pre-limit, as $\epsilon \downarrow 0$.

**Proposition 11.** *Let $\hat{q}^* \in \widehat{\mathcal{Q}}^*(M)$ for some admissible menu $M \in \widehat{\mathcal{M}}$ in heavy traffic. Then, there exists a sequence of strategy profiles $(q^{(\epsilon)})_{\epsilon>0}$ with corresponding steady-state waiting times $W^{(\epsilon)} = \{W_i^{(\epsilon)}(q^{(\epsilon)})\}_{i\in[n]}$ and matching probabilities $p^{(\epsilon)} = [p_{ij}^{(\epsilon)}(q^{(\epsilon)})]_{i\in[n],j\in[m]}$ such that $(q^{(\epsilon)}, W^{(\epsilon)}, p^{(\epsilon)})$ is a $\Delta^{(\epsilon)}$-equilibrium profile for a sequence $(\Delta^{(\epsilon)})_{\epsilon>0}$ that satisfies $\lim_{\epsilon\downarrow 0} \Delta^{(\epsilon)} = 0$.*

**Remark 3.** A possible shortcoming of the definition of a heavy traffic equilibrium in Definition 12 is that the sequence of strategy profiles $\{q^{(\epsilon)}\}_{\epsilon>0}$ that defines a heavy traffic equilibrium is not required to be a sequence of pre-limit equilibria. Thus, it is possible that a heavy traffic equilibrium is not the limit of any sequence of pre-limit equilibria. Proposition 11, however, guarantees that the strategy profiles $\{q^{(\epsilon)}\}_{\epsilon>0}$ are $\epsilon$-equilibria in the pre-limit and so the incentives that customers have to deviate from the strategy $q^{(\epsilon)}$ become negligible as $\epsilon \downarrow 0$. $\diamond$

The definition of a heavy-traffic equilibrium highlights an important feature of our asymptotic analysis of an equilibrium. Namely, to characterize a heavy traffic equilibrium it is not enough to identify the limiting strategy $\hat{q}^*$ but we must also specify the direction $\hat{\phi}^*$ of convergence. The reason is that the limiting vector of steady-state waiting times $\widehat{W}^*$ is not just a function of $\hat{q}^*$but also of $\hat{\phi}^*$. We illustrate this point with the following example.

**Example 1.** *Consider the system in Figure 3.4 with two customer types ($|\Theta| = 2$), two servers ($m = 2$) and two service classes ($n = 2$) each served exclusively by one of the servers. The arrival and service rates in the $\epsilon^{th}$ system are given by $\alpha^{(\epsilon)} = A - a\,\epsilon = (2, 1) - (1, 0)\,\epsilon$ and $\mu = (\mu_1, \mu_2) = (1, 2)$, respectively. Customers of type 1 prefer server 1 over server 2 (i.e., $V_{11} > V_{12}$) while the opposite is true for customers type 2 (i.e., $V_{21} < V_{22}$).*

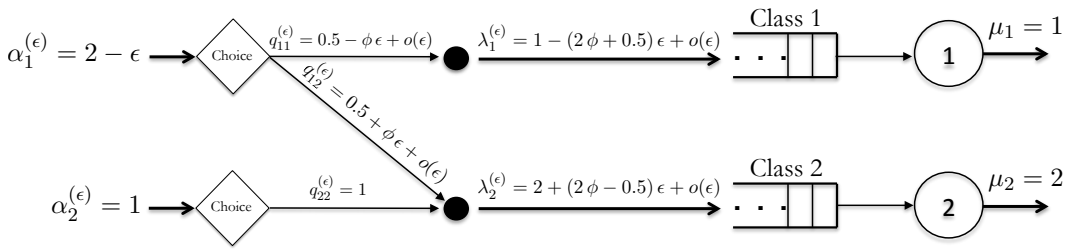



Figure 3.4: Example with two customer types, two service classes and two servers.

*For the given values of the arrival and service rates as well as the preferences of the customers, it should be intuitively clear that an equilibrium strategy $q^{(\epsilon)}$ for the $\epsilon^{th}$ system takes the form $q_{22}^{(\epsilon)} = 1 - q_{21}^{(\epsilon)} = 1$ and $q_{12}^{(\epsilon)} = 1 - q_{11}^{(\epsilon)} = 0.5 + \phi\,\epsilon + o(\epsilon)$ for some scalar $\phi$ such that $|\phi| < 1/4$ (this condition ensures that the queueing system is stable for $\epsilon >$ small enough). The strategy profile $q^{(\epsilon)}$ converges, as $\epsilon \downarrow 0$, to $\hat{q}^*$ given by $\hat{q}_{11}^* = \hat{q}_{12}^* = 1/2$ and $\hat{q}_{22}^* = 1 - \hat{q}_{21}^* = 1$. Thus, in the limit, half of type 1 customers are served by server 1 and all type 2 customers are served by server 2.*

*Since each service class behaves as an M/M/1 queue, the scaled steady-state waiting times in the $\epsilon^{th}$ system are given by*

$$\widehat{W_1}^{(\epsilon)}(q^{(\epsilon)}) = \frac{1}{0.5 + 2\,\phi + O(\epsilon)} \qquad \textit{and} \qquad \widehat{W_2}^{(\epsilon)}(q^{(\epsilon)}) = \frac{1}{0.5 - 2\,\phi + O(\epsilon)}.$$

*It follows from the above that to characterize the limiting value of the waiting times the limiting strategy $\hat{q}^*$ is not enough, and the direction $\phi$ of convergence of the strategy profile $q^{(\epsilon)}$ is necessary. To pinpoint the precise value of $\phi$ that will ensure that $\hat{q}^*$ is a heavy traffic equilibrium we must impose the best-response condition in Definition 12. In this example, customers type 1 randomize between service classes 1 and 2 and so they must be indifferent between them. It follows that*

$$\lim_{\epsilon \downarrow 0} \left( \widehat{W_1}^{(\epsilon)}(q^{(\epsilon)}) - \widehat{W_2}^{(\epsilon)}(q^{(\epsilon)}) \right) = \frac{V_{11} - V_{12}}{\delta}.$$

*Letting* $\beta := (V_{11} - V_{12})/\delta$, *we get that choosing*

$$\phi^* = \frac{2 - \sqrt{4 + \beta^2}}{4\,\beta}$$

*ensures that* $\hat{q}^*$ *is indeed a heavy traffic equilibrium in the sense of Definition 12.* □

**3.4.4 Pareto Improvement and Chained DAGs** Consider an admissible menu in heavy traffic $M \in \widehat{\mathcal{M}}$ with a heavy traffic equilibrium $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$ and let $\mathbb{C} = \{\mathbb{C}_1, , \ldots, \mathbb{C}_K\}$ be its corresponding collection of CRP components[5]. Our next result shows that under fairly general conditions we can always find another menu with a heavy traffic equilibrium with the same collection of CRP components that (weakly) Pareto dominates $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$.

**Proposition 12.** *Consider an admissible menu* $M \in \widehat{\mathcal{M}}$ *with a heavy traffic equilibrium* $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$ *and CRP components* $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_K\}$. *Denote by* $\widehat{W}_{\mathbb{C}_k}$ *the limiting scaled waiting time of component* $\mathbb{C}_k$ *for* $k \in [K]$ *and assume (after relabeling if necessary) that* $\widehat{W}_{\mathbb{C}_1} \leq \widehat{W}_{\mathbb{C}_2} \leq \cdots \leq \widehat{W}_{\mathbb{C}_K}$. *Suppose that* $1/|a| \leq \widehat{W}_{\mathbb{C}_1} < \widehat{W}_{\mathbb{C}_2}$, *then there exists a menu* $M' \in \widehat{\mathcal{M}}$ *with a heavy traffic equilibrium* $(\hat{q}^*, \widehat{W}', \hat{p}^*)$ *with the same set of CRP components* $\mathbb{C}$ *and such that* $\widehat{W}' \leq \widehat{W}^*$. *Furthermore, in this new equilibrium the CRP components in* $\mathbb{C}$ *are connected through a chained DAG (see Definition 7).*

Proposition 12 is significant as it reveals that for the purpose of finding an optimal service menu we can essentially restrict ourselves to menus that induce a heavy traffic equilibrium with CRP components connected by a chained DAG. We will take full advantage of this property in Section 3.7, where we study the class of Partition service menus. We also note that we can extend the result in the proposition to include the degenerate case $1/|a| < \widehat{W}_{\mathbb{C}_1} = \widehat{W}_{\mathbb{C}_2}$[6]. In this case, however, we can only show that for any $\varepsilon > 0$ (small) there

5. CRP components were defined in Definition 3

6. Recall that by Proposition 4 the case $1/|a| = \widehat{W}_{\mathbb{C}_1} = \widehat{W}_{\mathbb{C}_2}$ is not possible.

exists a $\varepsilon$-heavy-traffic equilibria with the same CRP components connected by a chained DAG that (weakly) Pareto dominates $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$.

## 3.5 Service Menus with Two Servers

In this section, we return to the two server example to illustrate the analysis of heavy traffic equilibria. In this setting, we are able to obtain a complete solution and derive a number of insights that we will use later to analyze the general case with an arbitrary number of servers.

Before we begin studying the possible menus and their equilibria, it helps to establish some *benchmarks* for what performance one might aim for along the dimensions of average waiting time and matching reward, respectively. Looking first at average waiting time, it is quite straightforward to see that one can not expect an average delay smaller than that of a single server queue with service rate equal to the total service rates of the $m$ servers, and the arrival rate equal to the total arrival rate of the customer types. Under heavy-traffic, we denote this ideal scaled delay as $\overline{W}_{\min}$:

$$\overline{W}_{\min} = \frac{1}{|a|}. \tag{3.5.1}$$

(Recall that $|a| = \sum_{\theta \in \Theta} a_\theta$ is the aggregated capacity slack.) Next, turning to matching reward, the following max-flow linear program solves the matching that a central planner would like to implement if she had complete control over the assignment of customers to servers and were only concerned with maximizing matching rewards.

$$\overline{V}_{\max} := \max_{f_{\theta j} \geq 0} \sum_{\theta, j} f_{\theta j} V_{\theta j} \quad \text{subject to} \quad \sum_{j} f_{\theta j} = A_\theta, \ \forall \theta \in \Theta \quad \text{and} \quad \sum_{\theta} f_{\theta j} = \mu_j, \ j = 1, 2. \tag{3.5.2}$$

It thus follows that $\overline{V}_{\max}$ is an upper bound on the cumulative matching value that can be achieved by any menu in equilibrium.

We now consider the space of admissible menus. Recall from Section 3.3 that with two servers, there are three possible service classes, namely, Class 1 served only by server 1, Class 2 served only by server 2, and Class 3 served by both servers. All five of the service menus we discussed in Section 3.3 are admissible in heavy traffic. These menus are:

- DEDICATED MENU (D), in which Classes 1 and 2 are offered,
- SINGLE-LINE MENU (SL), in which only service Class 3 is offered,
- FULL MENU (F), in which all three classes are offered,
- $N_i$ MENU, in which Classes $i$ and 3 are both offered, for $i = 1, 2$.

Also recall that when there are only two servers, we can index the customer types according to their relative preferences over the two servers. Specifically, we order the customer types $\{\theta_1, \theta_2, \ldots, \theta_{|\Theta|}\}$ such that $\Delta V_{\theta_i} \leq \Delta V_{\theta_j}$ for all $1 \leq i < j \leq |\Theta|$, where $\Delta V_{\theta_i} = V_{\theta 2} - V_{\theta_1}$. Let us define the subsets $\Theta_0 := \{\theta \in [\Theta] \colon \Delta V_\theta = 0\}$, $\Theta_1 := \{\theta \in [\Theta] \colon \Delta V_\theta < 0\}$, $\Theta_2 := \{\theta \in [\Theta] \colon \Delta V_\theta > 0\}$ so that customers in $\Theta_0$ are indifferent between the two servers while customers in $\Theta_i$ strictly prefer server $i = 1, 2$. We also define $A_i := \sum_{\theta \in \Theta_i} A_\theta$ to be the limiting arrival rate of customers in $\Theta_i$, for $i = 0, 1, 2$.

To fix ideas and notation, let us assume that the capacity of server 1 is insufficient to serve all customers who strictly prefer server 1 over server 2, i.e., $A_1 > \mu_1$. The case $A_2 > \mu_2$ is of course equivalent after relabeling the servers. The case $A_i < \mu_i$ for $i = 1, 2$ is discussed at the end of this section in Remark 4. Finally, the boundary case $A_i = \mu_i$ for $i = 1, 2$ can be analyzed using similar ideas and, for brevity, is omitted.

Under the assumption $A_1 > \mu_1$, Figure 3.5 depicts an example of the performance of the heavy traffic equilibrium of the five menus in the delay vs. reward quadrant. As we can see from the figure, we can split the five menus into two groups:

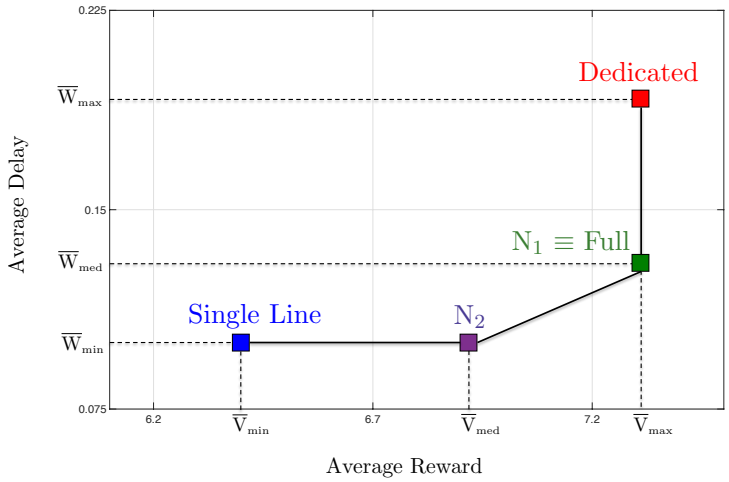



Figure 3.5: Performance of the heavy traffic equilibrium for Dedicated, Single Line, Full, $N_1$ and $N_2$ menus. DATA: $|\Theta| = 5$, $A = (1, 1, 3, 3, 2)$, $a = (2, 2, 2, 2, 2)$, $V_{\cdot 1} = (10, 10, 5.1, 9, 2)$, $\delta = 1$, $V_{\cdot 2} = (2, 8, 5, 10, 4)$ and $\mu_1 = 3$, $\mu_2 = 7$.

1. **Delay Minimizing Menus:** The Single Line and $N_2$ menus achieve the minimum possible average scaled waiting time, $\overline{W}_{\min}$.

2. **Reward Maximizing Menus:** The Dedicated, Full and $N_1$ menus all lead to equilibria that attain maximum possible matching reward, $\overline{V}_{\max}$. Furthermore, the equilibrium of the Full and $N_1$ turn out to be equivalent in heavy traffic.

To get some intuition about this segmentation of the menus, consider Figure 3.6 that summarizes the heavy traffic equilibrium outcome for the five menus in terms of matching flows and corresponding DAG of CRP components. Note that both the Single Line and the $N_2$ menu induce a single CRP component in equilibrium. For the Single Line this is trivially the case and for the $N_2$ menu this follows from the fact that $A_1 > \mu_1$ and so there are enough customers who want to join class 3 to ensure a positive flow from class 3 to server 2 in equilibrium. Thus, with a single CRP component, Proposition 4 implies that customers' average waiting time is minimized and equals $\overline{W}_{\min}$. In terms of matching rewards, however, the Single Line and $N_2$ menus have different performance. On one hand, the Single Line

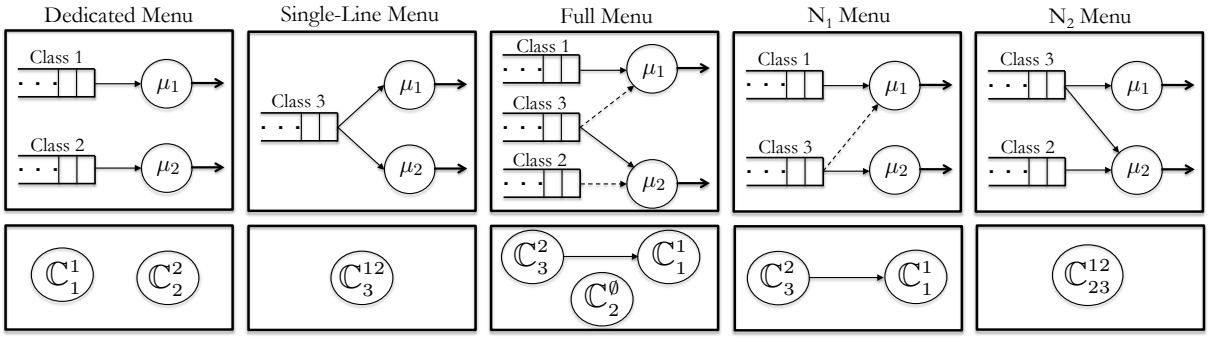



Figure 3.6: Summary of the heavy traffic equilibrium outcome for the five menus in terms of matching flows and DAG of CRP components.
**Top Panel** depicts the matching flows between the customer types and service classes indicate equilibrium strategies. Solid (dashed) arrows between the service classes and servers indicate asymptotically non-negligible (negligible) flows.
**Bottom Panel** depicts the DAG that emerges in the heavy traffic equilibrium, where $\mathbb{C}_{\mathcal{C}}^{\mathcal{S}}$ denotes a CRP that includes service classes in $\mathcal{C}$ and servers in $\mathcal{S}$.

produces the lowest average reward ($\overline{V}_{\min}$ in Figure 3.5) among all five menus, while the $N_2$ menu generates an intermediate reward value $\overline{V}_{\text{med}}$ in Figure 3.5. To compute the value of $\overline{V}_{\min}$ note that in the Single Line menu all customers –irrespective of their type– are matched to servers in proportion to their service rate, that is,

$$\overline{V}_{\min} = \sum_{\theta \in \Theta} \frac{A_\theta}{|A|} \left( \frac{\mu_1}{\mu_1 + \mu_2} V_{\theta 1} + \frac{\mu_2}{\mu_1 + \mu_2} V_{\theta 2} \right). \tag{3.5.3}$$

On the other hand, to compute $\overline{V}_{\text{med}}$, we note that since the $N_2$ menu induces a single CRP component, the two service classes offered in $N_2$ (namely, classes 2 and 3) have the same waiting time in equilibrium. As a result, all customer types that strictly prefer server 1 join class 3 and all customer types that strictly prefer server 2 join class 2. Customers who are indifferent between the two servers are also indifferent between the two service classes since they have the same waiting time. However, the assumption $A_1 > \mu_1$ together with our 'optimistic' formulation (see Remark 1) imply that all these indifferent customers join class

2 in equilibrium. It follows that

$$\overline{V}_{\text{med}} = \sum_{\theta \in \Theta_1} \frac{A_\theta}{|A|} \left( \frac{\mu_1}{A_1} V_{\theta 1} + \frac{A_1 - \mu_1}{A_1} V_{\theta 2} \right) + \sum_{\theta \in \Theta_0 \cup \Theta_2} \frac{A_\theta}{|A|} V_{\theta 2}. \tag{3.5.4}$$

Let us turn to the three reward maximizing menus: Dedicated, Full and $N_1$. The common feature of these three menus is that they all include service Class 1 and since $A_1 > \mu_1$: *(i)* server 1 exclusively serves customer types that prefer it the most, leading to reward maximization, and *(ii)* there is at least one customer type in $\Theta_1$ that must be indifferent between joining Class 1 and some other class. It is precisely this indifference condition that pinpoints the heavy traffic equilibrium for these three menus. In terms of the average delay experienced by customers in equilibrium, the Full and $N_1$ menus produce the same average delay $\overline{W}_{\text{med}}$, while the Dedicated produces an average delay $\overline{W}_{\text{max}}$, with $\overline{W}_{\text{med}} \leq \overline{W}_{\text{max}}$. This is an example of the Pareto improvement described in Proposition 12, since the DAGs of Full and $N_1$ menu can be seen as chaining the CRP components of the Dedicated menu.

The following proposition summarizes the performance of the heavy traffic equilibrium for each of the five menus.

**Proposition 13.** *Suppose that $A_1 \geq \mu_1$. Then, the performance of the heavy traffic equilibrium, in terms of average waiting times and matching rewards, for each of the five menus is given by:*

| | Dedicated | $N_1$ | Full | $N_2$ | Single-line |
|---|---|---|---|---|---|
| Avg. Waiting Times | $\overline{W}_{\text{max}}$ | $\overline{W}_{\text{med}}$ | $\overline{W}_{\text{med}}$ | $\overline{W}_{\text{min}}$ | $\overline{W}_{\text{min}}$ |
| Avg. Matching Rewards | $\overline{V}_{\text{max}}$ | $\overline{V}_{\text{max}}$ | $\overline{V}_{\text{max}}$ | $\overline{V}_{\text{med}}$ | $\overline{V}_{\text{min}}$ |

Table 3.1: Menu outcomes for the two-server case.

*such that $\overline{W}_{min} \leq \overline{W}_{med} \leq \overline{W}_{max}$ and $\overline{V}_{min} \leq \overline{V}_{med} \leq \overline{V}_{max}$. The values of $\overline{W}_{min}$, $\overline{V}_{min}$,*

*$\overline{V}_{med}$ and $\overline{V}_{max}$ are derived in equations* (3.5.1)-(3.5.2) *and the values of $\overline{W}_{med}$ and $\overline{W}_{max}$ are derived in the proof of the proposition in equations* (3.A3) *and* (3.A2)*, respectively.*

Let us conclude this section with the following two remarks.

**Remark 4.** (First Best Menu). *There are two cases in which the service provider can achieve a first outcome, namely, $\overline{W}_{min}$ delays and $\overline{V}_{max}$ rewards:*

(i) *Suppose $A_i = \mu_i$ for either $i = 1$ or $i = 2$. Then, offering the $N_{3-i}$ menu achieves first best. To see this, take for example the case $A_1 = \mu_1$, then we get from* (3.5.4) *that $\overline{V}_{med} = \overline{V}_{max}$ and from Proposition 13 we conclude that the $N_2$ menu Pareto dominates the other four menus as it achieves the best performance in both dimensions (waiting times and rewards).*

(ii) *Consider the case $A_i < \mu_i$ for $i = 1, 2$, that is, when both servers have excess capacity to serve the customer types that strictly preferred them. In this case, the Full menu is optimal as it Pareto dominates the other four menus. To see this, note that the condition $A_i < \mu_i$ implies that a stable strategy is to have customers in $\Theta_i$ joining class $i$ (for $i = 1, 2$) and the indifferent customers in $\Theta_0$ joining class 3. This strategy will naturally maximize average matching rewards. Furthermore, in the heavy traffic regime, this strategy will induce a single CRP component and so the average waiting time of each service class is the same. Thus, no customer type has an incentive to switch to another class. As a result, a Full menu achieves simultaneously the minimum average waiting time and the maximum matching reward and it is therefore optimal.*

*We also note that the equivalence between the Full and $N_1$ menus does not hold anymore when $A_i < \mu_i$ for $i = 1, 2$. In this case, the $N_1$ menu does not produce maximum matching rewards since some customers in $\Theta_2$ will have to be served by server 1 in equilibrium.* $\diamond$

**Remark 5.** (Trivial CRP Components) *In Figure 3.6, the DAG induced by the Full menu has a CRP component* $\mathbb{C}_2^{\emptyset}$ *which includes class 2 and no server. This anomaly happens because even though class 2 is offered there are no customers joining this class in the heavy traffic equilibrium. Note that despite the fact that there is no flow of customers joining class 2, we still need to assign a waiting time to this class to enforce equilibrium conditions. Chapter 2 contains a detailed discussion of how to compute the waiting time of these trivial CRP components in heavy traffic.* $\diamond$

## 3.6 First Best Menus

We saw in our discussion of the two-server case that it is sometimes possible to offer a service menu that achieves a first best outcome, that is, maximum possible matching values and minimum possible average waiting times simultaneously (see Remark 4). In this section, we investigate conditions under which a first best menu exists in a system with an arbitrary number of servers. To this end, we find it convenient to first discuss two special menus, namely the Single Line (SL) and Dedicated (D) menus, which exhibit extreme and contrasting performance in terms of matching rewards and waiting times. While a Single Line menu minimizes waiting times at the expense of matching values the opposite is true for the Dedicated menu. This is illustrated in Figure 3.5 for the two-server case.

**3.6.1 Single Line Menu:** In the Single Line menu the service provider offers a single service class which is served by all servers. This is the simplest and most common service configuration used in practice in which all $m$ servers serve a single service class. By Proposition 4, the Single Line exhibits complete resource pooling and therefore minimizes average waiting times in heavy traffic, $\widehat{W}^{\text{SL}} = 1/|a|$. Thus, it is an optimal menu when the service provider is interested in minimizing customers' average waiting times exclusively (i.e., $\zeta = \infty$). However, as we saw in the two-server model, the Single-Line menu is not

Pareto optimal in general. Actually, our next result reveals that while the Single-Line menu minimizes waiting times, it also minimizes average matching rewards.

**Theorem 4.** *For any an admissible menu in heavy traffic $M \in \widehat{\mathcal{M}}$ and any heavy traffic equilibrium strategy profile $\hat{q}^* \in \widehat{\mathcal{Q}}^*(M)$ under $M$, let $\overline{V}(M, \hat{q}^*)$ be the average matching rewards under the pair $(M, \hat{q}^*)$. Let also $\overline{V}^{\mathrm{SL}}$ be the average matching reward under the Single Line menu. Then,*

$$\overline{V}^{\mathrm{SL}} \leq \overline{V}(M, \hat{q}^*).$$

The proof of Theorem 4 can be found in Section 3.A.5. Intuitively, the key limitation of the Single Line menu exposed in Theorem 4 is its inability to customize the matching between customers and servers since all customers are essentially treated equally. This raises the question of how to design a menu that maximizes customer's rewards among all menus that have an equilibrium with a single CRP component. We will return to this question in Section 3.8.2.

**3.6.2 Dedicated Menu:** In the Dedicated menu each server operates independently serving its own service class. In other words, the matching topology $M^{\mathrm{D}} = [m^{\mathrm{D}}_{ij}] \in \{0,1\}^{m \times m}$ of the Dedicated menu satisfies $m^{\mathrm{D}}_{ij} = \mathbb{1}(i = j)$. In contrast to the Single Line menu, the Dedicated menu has no resource pooling but offers full flexibility to match customers to servers, and in Theorem 5 below we show that this matching flexibility is actually maximal. To this end, let us consider the following max-flow problem for system $\epsilon$, which is central to

our characterization of first best menus:

$$\begin{aligned}
\overline{\mathcal{V}}^{(\epsilon)} := \max_{f_{\theta j}^{(\epsilon)} \geq 0} \ & \sum_{\theta,j} f_{\theta j}^{(\epsilon)} V_{\theta j} && && \textbf{(Max-flow)} \\
\text{subject to} \quad & \sum_{j} f_{\theta j}^{(\epsilon)} = \alpha_{\theta}^{(\epsilon)} && \forall \theta \in [\Theta], && \text{(flow balance)} \\
& \sum_{\theta} f_{\theta j}^{(\epsilon)} \leq \mu_j && \forall j \in [m], && \text{(capacity)}
\end{aligned}$$

where $f_{\theta j}^{(\epsilon)}$ represents the flow of customers type $\theta$ served by server $j$. We note that the value of $\overline{\mathcal{V}}^{(\epsilon)}$ corresponds to the maximum average matching reward that a central planner can achieve if she has full control on how to match customers to servers. It follows that $\overline{\mathcal{V}}^{(\epsilon)}$ provides an upper bound on the maximum matching reward that the service provide can get from any equilibrium. Interestingly, our next result shows the Dedicated menu achieves this upper bound asymptotically. In other words, the Dedicate menu is an optimal menu if the service provider is completely insensitive to waiting times (i.e., $\zeta = 0$). This is interesting because customers' equilibrium strategies still depend on the waiting times of each service class, and so even if $\zeta = 0$ the service provider cannot simply disregard the effect of waiting times on the overall performance.

**Theorem 5.** *Let $V^{(\epsilon)}$ be the matching value of an equilibrium for the Dedicated menu for system $\epsilon$. Then, $\overline{\mathcal{V}}^{(\epsilon)} - V^{(\epsilon)} = \mathcal{O}(\epsilon)$, i.e., the* Dedicated *menu asymptotically maximizes average matching value in heavy traffic.*

Proof Sketch: (See Section 3.A.6 for a complete proof) Since some elements of the proof are quite insightful and useful for the discussion that follows, we provide a quick proof sketch here and defer a full version to the Appendix. First, let us introduce the dual variables $\eta_{\theta}^{(\epsilon)}$ for the flow balance for customer type $\theta$, and $\omega_{j}^{(\epsilon)}$ for the capacity constraint for server $j$. The

dual problem to ($\mathbf{Max\text{-}flow}$) is

$$\min_{\omega_j^{(\epsilon)} \geq 0, \eta_\theta^{(\epsilon)}} \quad \sum_\theta \alpha_\theta^{(\epsilon)} \eta_\theta^{(\epsilon)} + \sum_j \mu_j \, \omega_j^{(\epsilon)} \quad \text{subject to} \quad \eta_\theta^{(\epsilon)} + \omega_j^{(\epsilon)} \geq V_{\theta j} \qquad \forall \theta, j. \tag{Dual-Max-flow}$$

The main idea is to show that any equilibrium strategy $q^{(\epsilon)} \in \mathcal{Q}^*(M^{\mathrm{D}})$ for the Dedicated menu (which exists due to Theorem 3) induces a vector of flow rates from customers to servers, $f_{\theta j}^{(\epsilon)}(q^{(\epsilon)})$, that can be used to construct a feasible dual solution such that approximate complementary slackness holds in the following sense:

$$\left(\mu_j - \sum_\theta f_{\theta j}^{(\epsilon)}\right) \omega_j^{(\epsilon)} = \mathcal{O}(\epsilon), \qquad \text{and} \qquad \left(\eta_\theta^{(\epsilon)} + \omega_j^{(\epsilon)} - V_{\theta j}\right) f_{\theta j}^{(\epsilon)} = 0,$$

for all $\theta$ and $j$, which then guarantees that $f_{\theta j}^{(\epsilon)}$ is approximately optimal for ($\mathbf{Max\text{-}flow}$) with an $\mathcal{O}(\epsilon)$ additive suboptimality, which vanishes as $\epsilon \downarrow 0$.

In particular, given that the expected waiting time at queue $j$ under $f_{\theta j}^{(\epsilon)}$ equals $\widehat{W}_j^{(\epsilon)} = 1/(\mu_j - \sum_\theta f_{\theta j}^{(\epsilon)})$, we define for all $j \in [m]$ and $\theta \in \Theta$,

$$\omega_j{}^{(\epsilon)} = \delta \, \widehat{W}_j^{(\epsilon)} \qquad \text{and} \qquad \eta_\theta{}^{(\epsilon)} = \max_j \left\{ V_{\theta j} - \omega_j{}^{(\epsilon)} \right\} \tag{3.6.1}$$

as a feasible dual solution. By the definition above, $\eta_\theta^{(\epsilon)}$ is in fact the utility of type $\theta$ customers. To see the intuition behind why complementary slackness holds, the first set of conditions follow from the definition of $\omega_j^{(\epsilon)}$. For the second set, since under any equilibrium, $f_{\theta j}^{(\epsilon)} > 0$ only if $V_{\theta j} - \omega_j^{(\epsilon)} \geq \eta_\theta^{(\epsilon)}$, exact complementary slackness holds for the second set of conditions. $\square$

**Remark 6.** *As we alluded to in the proof sketch, the optimal dual variables of* ($\mathbf{Max\text{-}flow}$) *(for the limiting case $\epsilon = 0$) have the following interpretation: $\omega_j$ denotes the limiting scaled mean delay disutility for server $j$ under the Dedicated menu, and $\eta_\theta$ denotes the average*

*utility of customer type $\theta$ under delay disutilities $\{\omega_j\}$. However, the dual solution is only determined up to a translation; for any $\tau$, $(\eta_\theta + \tau)$ and $(\omega_j - \tau)$ are also an optimal dual solution. Therefore without loss of generality, we can assume $\min_j \omega_j = 0$, so that the true scaled delay disutility for server $j$ is $\delta\widehat{W}_j = \omega_j + \omega_0$ for some $\omega_0$.*

Our next somewhat surprising result generalizes Theorem 5 in the sense that any menu $M$ which includes the Dedicated menu as a sub-menu also maximizes the average matching value.

**Theorem 6.** *Let $M$ be any service menu which includes the Dedicated menu as a submenu. That is, for every server $j$, there exists a service class $i(j)$ such that $m_{i(j),j} = 1$, and $m_{i(j),j'} = 0$ for any $j' \neq j$. Then the menu $M$ attains the maximum matching reward under any heavy-traffic equilibrium.*

The detailed proof of Theorem 6 can be found in Section 3.A.7. The crux of the proof relies on two observations. First, if there is a service class $i$ and server $j$ with $\hat{p}^*_{ij} > 0$ then $\widehat{W}_i$, the limiting scaled mean delay of service class $i$, equals $\widehat{W}_j$, the limiting scaled mean delay of the service class dedicated to server $j$. This $\widehat{W}_j$ still gives the dual $\omega_j$ as in the proof sketch above. Second, if a type $\theta$ joins a service class $i$ which has $\hat{p}^*_{ij} > 0$ and $\hat{p}^*_{ij'} > 0$ for servers $j \neq j'$ then $V_{\theta j} = V_{\theta j'}$ otherwise type $\theta$ has a strictly improving deviation to a dedicated service class. These two can be combined to get back the condition that the limiting flow from type $\theta$ to server $j$, $f_{\theta j} > 0$, only if $V_{\theta j} - \delta\widehat{W}_j = \eta_\theta = \max_{j'}\{V_{\theta j'} - \omega_{j'}\}$.

### 3.6.3 Necessary and Sufficient Conditions for First Best Outcomes:

From the performance of the Single Line and Dedicated menus we have that for a menu to achieve first best it must simultaneously induce an equilibrium with (i) a single CRP component and (ii) matching flows that coincide with the solution of the (**Max-flow**) problem. In this section, we use this insight to identify necessary and sufficient conditions for a first best outcome to be achievable.

**Theorem 7.** (Necessary Conditions) *If the service provider is able to achieve a first best outcome, then there exists a solution to* (**Max-flow**) *with* $\epsilon = 0$ *such that the following two conditions hold:*

1. *The arcs associated with strictly positive flows form a connected graph.*
2. *Every customer type weakly prefers their matching outcome to that of any other customer type.*

A proof of this theorem can be found in Section 3.A.8, but we will briefly provide some intuition here. If a first best outcome can be achieved, then the flows between service classes and servers must form a connected graph to support a single CRP component. This implies that the flows between customer types and servers must also form a connected graph. Similarly, since a first best outcome necessarily achieves the maximum possible matching values, we know that the flows between customer types and servers (via the service classes offered) must also be a solution to (**Max-flow**). Since the flows form an equilibrium, we know that no customer type prefers the matching outcome of any other customer type. Note that we do not need to consider waiting times when evaluating the incentive compatibility of the outcome as when there is a single CRP component, all service classes have the same expected delay.

A particular system will fail the first condition when the only solutions to (**Max-flow**) have some subset of customer types exactly served by a particular subset of servers. In this case, the only solutions to (**Max-flow**) with $\epsilon = 0$ are disconnected. This could happen, for instance, if there is exactly enough service capacity to have all customer types served by their most preferred servers. We can only achieve a single CRP component in this situation is some customer types are served by servers that are not their most preferred server, meaning that it is not possible to achieve a single CRP component and a value maximising solution simultaneously.

The second condition will fail if it is not incentive compatible for the customers to maintain value maximising flows when expected delays are the same across service classes. This can happen because the solution to (**Max-flow**) is prioritising customer types with large differences between valuations. If customer type $\hat{\theta}$ and customer type $\tilde{\theta}$ both prefer server 1 to server 2, but $V_{\hat{\theta}1} - V_{\hat{\theta}2} > V_{\tilde{\theta}1} - V_{\tilde{\theta}2}$, then the (**Max-flow**) solution will allocate server 1 to $\hat{\theta}$ customers over $\tilde{\theta}$ customers. However, the customer types themselves care only that they prefer one matching outcome to another, and the degree to which they prefer one matching outcome does not matter. This can result in a mismatch between the solution to (**Max-flow**) and the customers preferences.

One extreme example of the second condition failing is when there exists $(\sigma_\theta)_{\theta\in[\Theta]}$ and $\{\beta_j\}_{j\in[m]}$ such that $V_{\theta j} = \sigma_\theta \beta_j$ for all $\theta \in [\Theta]$ and $j \in [m]$. In this case, no matter how the service classes are designed, all customers will have the same ordinal preferences over the matching outcomes of the different service classes. Hence it will be impossible to maintain a single CRP component in equilibrium when offering multiple service classes, as it will only be incentive compatible for different customer types to join different service classes if there are differences in expected delays between service classes. However, achieving a value maximising outcome will require that only customer types with higher values of $\sigma_\theta$ are served by servers with higher values of $\beta_j$.

Next, we provide sufficient conditions for a first best outcome to be achievable.

**Theorem 8.** (Sufficient Conditions) *The service provider is always able to achieve a first best outcome if there exists a solution to* (**Max-flow**) *with* $\epsilon = 0$ *such that the following two conditions hold:*

1. *The basic feasible activities induce a connected tree*
2. *Every customer type weakly prefers their matching outcome to that of any other customer type.*

*The menu that achieves the first best outcome is the menu in which there is a single service class for each customer type, and that service class consists of all of the servers that they are connected to in the Max Flow solution.*

As the proof of this theorem is short and intuitive, we include it here.

*Proof.* To see that these conditions are sufficient, we can consider what would happen if the proposed menu were offered. Since the flows form a connected tree, we know that these flows are those that would be achieved if this menu were to be offered, and each customer type were to join their assigned service class. As the resulting graph is connected, we know that a single CRP component is achieved, and hence minimum possible expected waiting times occur. Because each customer type weakly prefers their own matching outcomes to that of any other customer type, and waiting times across service classes are equal, it is an equilibrium for each customer type to join their assigned service class. □

This also provides some intuition as to why the conditions stated in Theorem 7 are not sufficient for a first best outcome to be possible. If the flows associated with the feasible activities form a tree, we can guarantee that there is a combination of menu and customer strategies will achieve these flows. If the flows associated with the feasible activities form a cycle, we cannot make the same guarantee.

We can satisfy the sufficient conditions by ensuring that different customer types have different rankings of servers. This will mean that it is incentive compatible for different customer types to join different service classes when the expected delays at the different service classes are the same. While there are different ways this can be achieved, we provide a some specific examples to help make the intuition clear.

One way that these sufficient conditions can be satisfied is to have 'enough' indifferent customers in the system. One example of how this could happen is if there is some mass of

customer types who have strict preferences between servers, and some mass of customers who have pairs of servers that they are indifferent between, but they prefer to all other servers. If there is enough service capacity so that all customer types with strict preferences can be served by their most preferred servers, and an ordering of servers so that for every pair of servers $(j, j+1)$ for $j = 1, ..., m-1$ there is a customer type who is indifferent between servers $j$ and $j+1$, then the sufficient conditions will be satisfied. The menu that achieves the first best outcomes offers a dedicated service class for each server, that is, a service class only served by this server, and an additional service class for each pair of servers $(j, j+1)$ for $j = 1, ..., m-1$. A single CRP component is achieved by have all customer types join the service class served by their most preferred pair of servers if they are indifferent between a pair of servers, and their most preferred server otherwise, and since every customer types is being served by their most preferred server or servers, it is both an equilibrium and a value maximising outcome.

Another way in which these conditions will be satisfied if every customer type has a different most preferred server, but the same second most preferred server. We additionally require that there is sufficient service capacity to serve each customer type by either their most preferred server or their second most preferred server, but there is insufficient service capacity to serve any customer type completely using their most preferred server. In this case we can achieve a first best outcome by designing a service class for each customer type, where the service class for type $\theta$ customers is served by both their most preferred server and their second most preferred server. If we offer this menu, since there is insufficient service capacity to serve any customer type completely using their most preferred server, we know a single CRP component will result if each customer type joins the service class designed for them. Additionally, since each customer type is joining the service class being served by their most preferred servers, it will be incentive compatible for them to do so. Thus an equilibrium and a value maximising outcome is also achieved.

These two examples demonstrate that putting a particular structure on the valuations themselves is insufficient for guaranteeing the existence of a first best outcome. It is the interaction between the valuations, arrival rates, and service capacity that tells us whether or not a first best outcome is achievable. We would also like to note that while the specific examples we mention here have all customer types being served by their most preferred or two most preferred servers, a first best outcome can sometimes be achieved with some customer types not being served at all by their most preferred server.

## 3.7 Partition Menus

In the previous section we identified conditions under which there exists a menu that achieves first best outcome. In general, however, first best cannot be achieved and an optimal menu must appropriately balance the trade-off between waiting times and matching rewards. In this section, we investigate this trade-off by restricting ourselves to the study of a special class of *Partition menus* in which the set of servers is partitioned into $K$ pools $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ for some $K \in [m]$. We will consider two classes of partition menus.

- Pure Partition Menus: These are menus in which each partition of servers $\mathcal{S}_k$ is dedicated to serving exclusively a single service class, say $\mathcal{C}_k$. The left panel in Figure 3.7 depicts an example of a pure partition menu with four customers classes and five servers. In this example, servers are partitioned into two sets $\mathcal{S}_1 = \{1, 2, 3\}$ and $\mathcal{S}_2 = \{4, 5\}$ with all servers in partition $\mathcal{S}_i$ serving exclusively service class $i$, for $i = 1, 2$.

  It is worth noticing that in heavy traffic, a partition menu consists of $K$ disconnected CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$, with $\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k)$.

- Chained Partition Menus: These are modified pure partition menus with some additional connectivity among the CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ so that

the underlying DAG has a chained structure (see Definition 7). Thus, every chained partition menu has associated an underlying pure partition menu that defines it. For example, the right panel in Figure 3.7 depicts a chained partition menu associated to the pure partition in the right panel that includes a link (dashed arc) connecting $\mathbb{C}_1$ to $\mathbb{C}_2$.

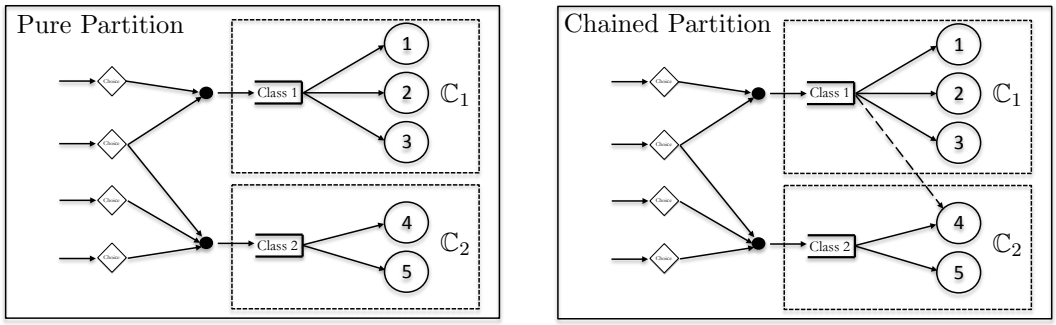



Figure 3.7: Example of pure partition and chained partition menus with two partitions of servers $\mathcal{S}_1 = \{1, 2, 3\}$ and $\mathcal{S}_2 = \{4, 5\}$.

While restrictive, partition menus have a number of desirable properties from a practical standpoint as they are easy to explain to customers and require limited scheduling coordination among the servers. For instance, in a pure partition menu, each server can manage FCFS requirements by tracking a single service class and customers only need to know their queueing position in a single line to assess their service status. In addition, by varying the number of partitions and their composition, partition menus offer a fair amount of flexibility that the service provider can use to trade-off matching rewards and waiting times. For instance, two notable examples of partition menus are the Single Line and the Dedicated menu discussed in Sections 3.6.1 and 3.6.2, respectively.

**3.7.1 Pure Partition Menus:** Let us fix a partition $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$, with all the servers in partition $\mathcal{S}_k$ serving a unique service class $\mathcal{C}_k$. It is easy to see that each pair $\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k)$ corresponds to a different CRP component in any heavy traffic equilibrium.

In this setting, a strategy profile can be represented by a matrix $q = [q_{\theta k}]$, where $q_{\theta k}$ is the probability that a type $\theta$ customer joins $\mathcal{C}_k$. Moreover, since each service class $\mathcal{C}_k$ is served exclusively by the servers in $\mathcal{S}_k$, the limiting matching probabilities $\hat{p}$ of any heavy traffic equilibrium must trivially satisfy $\hat{p}_{kj} = \mathbb{1}(j \in \mathcal{S}_k)\, \mu_j / \mu_{\mathcal{S}_k}$, where $\mu_{\mathcal{S}_k} = \sum_{j \in \mathcal{S}_k} \mu_j$. It follows that the average limiting reward that a type $\theta$ customer gets from joining service class $\mathcal{C}_k$ equals

$$\overline{V}_{\theta k} := \sum_{j \in \mathcal{S}_k} \frac{\mu_j \, V_{\theta j}}{\mu_{\mathcal{S}_k}}.$$

It is not hard to see that a pure partition menu with servers' partition $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ behaves, in the heavy traffic limit, as the Dedicated menu in which each partition of servers $\mathcal{S}_k$ acts as a '*super-server*' with capacity $\mu_{\mathcal{S}_k}$ and with a matrix of matching rewards $\overline{V} = [\overline{V}_{\theta k}]$ between customers types and super-servers. With this interpretation, on can show that Theorem 5 extends to this case in a relatively straightforward fashion. Specifically, consider the following modified version of the max-flow problem parameterized by the partition $\mathcal{S}$:

$$\overline{\mathcal{V}}_{\mathcal{S}}^{(\epsilon)} := \max_{f_{\theta k}^{(\epsilon)} \geq 0} \sum_{\theta, k} f_{\theta k}^{(\epsilon)} \, \overline{V}_{\theta k} \quad \text{subject to} \quad \sum_{k} f_{\theta k}^{(\epsilon)} = \alpha_{\theta}^{(\epsilon)} \quad \text{and} \quad \sum_{\theta} f_{\theta k}^{(\epsilon)} \leq \mu_{\mathcal{S}_k}, \tag{3.7.1}$$

where $f_{\theta k}^{(\epsilon)}$ represents the flow of customers type $\theta$ joining service class $\mathcal{C}_k$. As in the case of the Dedicated menu, $\overline{\mathcal{V}}_{\mathcal{S}}^{(\epsilon)}$ provides an upper bound on the maximum matching reward that the service provide can get from any equilibrium under the pure partition menu $\mathcal{S}$.

**Corollary 3.** *The average matching value $V_{\mathcal{S}}^{(\epsilon)}$ of any equilibrium for the pure partition menu with server partition $\mathcal{S}$ satisfies $\overline{\mathcal{V}}_{\mathcal{S}}^{(\epsilon)} - V_{\mathcal{S}}^{(\epsilon)} = \mathcal{O}(\epsilon)$.*

The proof the corollary can be found in Section 3.A.9. According to the previous result, all equilibria associated to the pure partition menu with partition $\mathcal{S}$ generate the same matching value $\mathcal{V}_{\mathcal{S}} := \min_{\epsilon \downarrow 0} \overline{\mathcal{V}}_{\mathcal{S}}^{(\epsilon)}$, in the heavy traffic limit. Under the following assumption on the max-flow problem (3.7.1), the limiting scaled waiting time of the pure partition menu

is also uniquely determined.

**Assumption 2.** *The solution to* (3.7.1) *with* $\epsilon = 0$ *is unique, and the basic feasible activities (that is, the edges* $(\theta, k)$ *with* $f_{\theta k} > 0$*) induce a connected tree.*

Assumption 2 is quite mild. For example if one were to generate a random instance of the service system by sampling the valuations $V_{\theta j}$ from non-atomic distributions then the maximum flow is unique with probability 1. Similarly, if either the arrival rates $A_\theta$ or the service rates $\mu_j$ are randomly sampled from non-atomic distributions then the maximum flow forest is a connected tree with probability 1.

Under Assumption 2, the limiting mean scaled waiting times of all service classes in the pure partition menu are determined up to an additive constant. This is because a customer type $\theta$ randomizing between service classes $\mathcal{C}_k$ and $\mathcal{C}_{k'}$ must be indifferent between them, and hence it must be true that $\overline{V}_{\theta k} - \delta \widehat{W}_k = \overline{V}_{\theta k'} - \delta \widehat{W}_{k'}$. The connectivity assumption then implies that knowing $\widehat{W}_k$ for some service class yields the waiting time for all service classes. Recall from Remark 6 that we can express the limiting scaled waiting times $\widehat{W}_k$ in terms of the dual variables $\omega_k$ for the service capacity constraints in (3.7.1). Specifically, there exist a vector of dual variables $\{\omega_k\}$ with $\min_k \omega_k = 0$ and a scalar $\omega_0$ such that $\delta \widehat{W}_k = \omega_k + \omega_0$. We use this representation in the next proposition to derive the precise waiting times under a partition menu.

**Proposition 14.** *Suppose Assumption 2 holds and let* $\{\omega_k\}$ *be a vector of dual variables for the service capacity constraints in* (3.7.1) *such that* $\min_k \omega_k = 0$*. Then, the limiting scaled mean waiting times for the pure partition menu are given by* $\widehat{W}_k^{\mathrm{PB}} = (\omega_k + \omega_0)/\delta$ *where* $\omega_0 \geq \delta/|a|$ *solves:*

$$\sum_{k=1}^{K} \frac{\delta}{\omega_k + \omega_0} = |a|.$$

We omit a formal proof as the intuition is simple: Under the pure partition menu, each service class $\mathcal{C}_k$ behaves asymptotically in the heavy traffic limit as an independent $M/M/1$

queue with service capacity $\mu_{\mathcal{S}_k}$. Thus a limiting scaled mean waiting time of $\widehat{W}_k$ implies $\lim_{\epsilon\downarrow 0}(\mu_{\mathcal{S}_k} - \lambda_k^{(\epsilon)})/\epsilon = 1/\widehat{W}_k$, where $\lambda_k^{(\epsilon)} = \sum_\theta f_{\theta k}^{(\epsilon)}$ is total arrival rate at service class $\mathcal{C}_k$. Further, $\lim_{\epsilon\downarrow 0}\sum_j(\mu_{\mathcal{S}_k} - \lambda_k^{(\epsilon)})/\epsilon = |a|$ by the heavy-traffic scaling in (3.4.1), which provides the necessary condition to pin down $\omega_0$.

**3.7.2 Chained Partition Menus:** Corollary 3 shows that pure partition menus maximize matching values for a given partition of servers. At the same time, they do not allow any form of capacity sharing between partitions, and this can lead to poor performance in terms of waiting times. To partially correct for this deficiency, we exploit the result Proposition 12 and consider a modified class of pure partition menus by "chaining" the service classes in increasing order of their waiting time. We refer to this class of menus as *chained* partitions. Intuitively, while a pure partition menu results in a DAG with $K$ disconnected CRP components (where each partition of servers along with their service class are in a CRP component of their own), a chained partition leads to a DAG which is a directed path (i.e., a single topological order), and allows for capacity pooling across CRP components. A special case of this construction is the $N_1$ menu in Section 3.5, where the chaining is apparent in Figure 3.6 (see also the right panel in Figure 3.7).

Recall that Proposition 3 provides a characterization of a class of scaled limiting waiting times that can be implemented in heavy traffic using a chained DAG. We take advantage of this result to derive the waiting times of a chained partition menu under the following additional assumption.

**Assumption 3.** *There exists an optimal vector* $\{\omega_k\}$ *of dual variables for the service capacity constraints in* (3.7.1) *such that* $0 = \widehat{\omega}_{(1)} < \widehat{\omega}_{(2)}$.

In the statement of the following proposition, we let $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ be a fixed partition of servers and $\mathcal{C}_k$ the service class connected to all servers in $\mathcal{S}_k$. We let $M_{\mathcal{S}}^{\text{PB}}$ denote the pure partition menu defined by $\{(\mathcal{C}_k, \mathcal{S}_k)\colon k \in [K]\}$.

**Proposition 15.** *Let Assumptions 2 and 3 hold and let* $\{\omega_k\}$ *be the optimal vector of dual variables satisfying the conditions in Assumption 3. Without loss of generality, let us relabel the* $K$ *service partitions in such a way that* $0 = \omega_1 < \omega_2 \leq \omega_3 \leq \cdots \leq \omega_K$. *Define a chained partition menu* $M_{\mathcal{S}}^{\text{CB}}$ *by extending the pure partition menu* $M_{\mathcal{S}}^{\text{PB}}$ *as follows: add a link connecting service class* $\mathcal{C}_k$ *to any server in partition* $\mathcal{S}_{k+1}$ *for* $k = 1, \ldots, K-1$. *The resulting chained partition menu generates maximum matching value* $\bar{\mathcal{V}}$ *and has limiting scaled waiting times given by*

$$\widehat{W}_1^{\text{CB}} = \frac{1}{|a|} \qquad \textit{and} \qquad \widehat{W}_k^{\text{CB}} = \frac{\omega_k}{\delta} + \frac{1}{|a|}, \quad k = 2, \ldots, K.$$

*It follows from Proposition 14 that* $\widehat{W}_k^{\text{CB}} = \widehat{W}_k^{\text{PB}} - \widehat{W}_1^{\text{PB}} + \frac{1}{|a|} \leq \widehat{W}_k^{\text{PB}}$. *Thus, from the prespective of the service provider, the chained menu* $M_{\mathcal{S}}^{\text{CB}}$ *(weakly) Pareto dominates the pure partition menu* $M_{\mathcal{S}}^{\text{PB}}$.

The proof of the proposition can be found in Section 3.A.10. We note that under the chained partition menu $\widehat{W}_1^{\text{CB}} = 1/|a|$, which by Proposition 4 is the delay under a completely pooled system and the lowest delay possible for a service class under any menu.

**3.7.3 Optimal Partitions:** We conclude this section by developing a mixed-integer linear program (MILP) to find an optimal chained partition menu. As these menus are constructed from partitions of servers, the number of possible menus grows rapidly with the number of servers. However, many of these menus will be Pareto dominated by others. The MILP formulation in Figure 3.8 assumes a fixed number $K$ of partitions and finds the optimal partition of servers $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$. By varying the value of $K$ from 1 to $m$ we can find the optimal chained partition menu. We will also describe a process that uses the MILP to identify the set of Pareto efficient chained menus.

The following are the main decision variables used in the MILP formulation:

-) $\{m_{kj},\ k \in [K], j \in [m]\}$: binary decision variables representing the service menu; 1 if server $j$ belongs to partition $\mathcal{S}_k$ and 0 otherwise.

-) $\{f_{\theta k},\ \theta \in [\Theta], k \in [K]\}$: flow of type-$\theta$ customers joining service class $\mathcal{C}_k$.

-) $\left\{f_{\theta kj},\ \theta \in [\Theta], k \in [K], j \in [m]\right\}$: flow of type-$\theta$ customers joining service class $\mathcal{C}_k$ and served by server $j$.

-) $\left\{\overline{V}_{\theta k},\ \theta \in [\Theta], k \in [K]\right\}$: average value that a type-$\theta$ customer gets from joining class $\mathcal{C}_k$.

-) $\left\{V_{\theta kj},\ \theta \in [\Theta], k \in [K], j \in [m]\right\}$: value that a type-$\theta$ customer gets from joining class $\mathcal{C}_k$ and service from server $j$. Desired behavior is that $V_{\theta kj} = \overline{V}_{\theta k}$ if $j \in \mathcal{S}_k$, and $V_{\theta kj} = 0$ otherwise.

-) $\{\omega_k,\ k \in [K]\}$: waiting time experienced by a customer who joins $\mathcal{C}_k$.

-) $\left\{\omega_{kj},\ k \in [K], j \in [m]\right\}$: scaled mean waiting time experienced by a customer who joins $\mathcal{C}_k$ and gets served by server $j$. Desired behavior is that $\omega_{kj} = \omega_k$ if $j \in \mathcal{S}_k$, and $\omega_{kj} = 0$ otherwise.

-) $m_{kj}$: 1 if server $j$ belongs to partition $\mathcal{S}_k$ and 0 otherwise.

-) $f_{\theta kj}$: flow of type-$\theta$ customers joining $\mathcal{C}_k$ and served by server $j$.

-) $f_{\theta k}$: flow of type-$\theta$ customers joining $\mathcal{C}_k$.

-) $V_{\theta kj}$: value that a type-$\theta$ customer gets from joining class $\mathcal{C}_k$ and service from server $j$.

-) $V_{\theta k}$: average value that a type-$\theta$ customer gets from joining class $\mathcal{C}_k$.

-) $\omega_{kj}$: waiting time experienced by a customer who joins $\mathcal{C}_k$ and get served by server $j$.

-) $\omega_k$: waiting time experienced by a customer who joins $\mathcal{C}_k$.

OBJECTIVE:

$$\mathcal{V}_K^{\mathrm{PB}} := \max \sum_{\theta kj} f_{\theta kj} \cdot V_{\theta j} - \zeta \sum_{kj} \mu_j \omega_{kj} \tag{3.7.2}$$

CONSTRAINTS:

**Server assignment:**
$$\sum_k m_{kj} = 1, \qquad \sum_j m_{kj} \geq 1. \tag{3.7.3}$$

**Enforcing max matching value:**
$$\eta_\theta + \delta\, \omega_k \geq \overline{V}_{\theta k}, \sum_{\theta kj} f_{\theta kj} \cdot V_{\theta j} = \sum_\theta A_\theta \eta_\theta + \delta \sum_{kj} \mu_j \omega_{kj}. \tag{3.7.4}$$

**Waiting time within partitions:**
$$\omega_k + (m_{kj} - 1)M \leq \omega_{kj} \leq \omega_k, \qquad \omega_{kj} \leq m_{kj} M. \tag{3.7.5}$$

**Customers' valuation for partitions:**
$$\overline{V}_{\theta k} + (m_{kj} - 1)M \leq V_{\theta kj} \leq \overline{V}_{\theta k}, \quad V_{\theta kj} \leq m_{kj} M, \tag{3.7.6a}$$

$$\sum_j \mu_j V_{\theta kj} = \sum_j m_{kj} \mu_j V_{\theta j}. \tag{3.7.6b}$$

**Flow balance:**
$$\sum_k f_{\theta k} = \alpha_\theta, \qquad \sum_{\theta k} f_{\theta kj} \leq \mu_j. \tag{3.7.7}$$

**Auxiliary flow constraints:**
$$f_{\theta kj} + (m_{kl} - 1)M \leq g_{\theta kjl} \leq f_{\theta kj}, \qquad g_{\theta kjl} \leq m_{kl} M, \tag{3.7.8a}$$

$$f_{\theta k} + (m_{kj} - 1)M \leq g_{\theta kj} \leq f_{\theta k}, \qquad g_{\theta kj} \leq m_{kj} M, \tag{3.7.8b}$$

$$\sum_l g_{\theta kjl} \cdot \mu_l = g_{\theta kj} \cdot \mu_j. \tag{3.7.8c}$$

**Non-negativity of decision variables**

$$\{f_{\theta k}\}, \{f_{\theta kj}\}, \{g_{\theta kj}\}, \{g_{\theta kjl}\}, \{V_{\theta k}\}, \{V_{\theta kj}\}, \{\omega_k\}, \{\omega_{kj}\} \geq 0 \quad \text{and} \quad \{m_{kj}\} \in \{0, 1\}. \tag{3.7.9}$$

Figure 3.8: MILP for finding the optimal partition menu with $K$ partitions.

The key idea in this MILP is that since a chained partition menu acts like a chained-dedicated menu on super-servers, we can use simultaneously the primal and dual constraints corresponding to max-flow problem in (3.7.1) to ensure that customer arrival rates are consistent with an equilibrium strategy profile. As mentioned in Remark 6, the dual variables $\omega_k$ can be interpreted as waiting times for the service classes. This means that by incorporating the dual constraints and dual variables into the MILP, we are able to include both the matching values and the waiting times of the service classes into the objective function. This is captured in the set of constraints (3.7.4).

For a given value of $\zeta$, we can solve the MILP to find the optimal maximizing chained partition menu. In addition, we can use the MILP to generate a Pareto frontier within the class of partition menus using standard multi-objective optimisation techniques for two objectives. We refer interested readers to Marler and Arora (2004) for a review of such methods.

## 3.8 Tailored Menus

In the previous section, we approached the problem of menu design by optimizing over the class of partitioned service menus in which each service class is associated with a unique and disjoint set of servers. In this section, we take an alternative perspective and use a *mechanism design* approach to tackle the problem of finding efficient menus. In particular, we use MILPs to design a service class or set of service classes for each customer type to optimize particular objectives. Since customers are acting strategically, we must also add constraints incentive compatibility constraints to the MILPs to ensure that the MILPs are optimizing over equilibrium outcomes. We refer to menus we design using this mechanism design approach as *Tailored* menus, as every service class is tailored to a particular customer type.

We use this mechanism design approach to design two different types of tailored menus. In Section 3.8.1, we formulate a MILP that identifies the delay-minimizing menu among the set of menus that achieve the maximum matching value outcome. In Section 3.8.2, we formulate a MILP that identifies the value-maximizing menu among the set of menus that achieve the minimum delay outcome, and have only a single service class for each customer type. A key advantage of tailored menus over partitioned menus is that they provide more flexibility to customize the matching between customer types and servers. On the flip side, tailored menus are more complex to design and possibly less practical from an implementation standpoint.

**3.8.1 Value Maximizing Tailored Menus:** The results in Sections 3.6.2 and 3.7.2 establish that a chained Dedicated menu maximizes the service provider's matching value. However, this menu has limited capacity pooling and therefore offers no guarantee of providing a good performance in terms of waiting times. To address this limitation we will formulate a MILP that minimizes waiting times over the class of menus that produce the maximum matching value.

Formally, we begin by solving (**Max-flow**) (under $\epsilon = 0$) to obtain the flow $\{f_{\theta j}\}$, which we will assume is unique and induces a connected tree by Assumption 2. Let $S_\theta := \{j : f_{\theta j} > 0\}$ denote the set of servers with non-zero flow from customer type $\theta$ in the maximum value flow. The menu design task then is to partition $S_\theta$ for each customer type $\theta$ into *service bundles* specifically intended for $\theta$.

We will use the following notation:

- $B_\theta = 2^{S_\theta} \setminus \emptyset$ denotes all the non-empty subsets of $S_\theta$.
- For type $\theta$, we call a set $b \in B_\theta$ a service bundle intended for type $\theta$, and also use it to denote the vector $b = (b_1, \ldots, b_m)$ where $b_j = 1$ if $j \in B$ and $b_j = 0$ otherwise. Note that although we associate bundle $b$ with a subset of servers, each such bundle is also

implicitly associated with a customer type. Thus we can have one subset of servers $S$ offered as two bundles, one for customer type $\theta$ and for customer type $\theta'$.

- $A_{\theta b} = \sum_j f_{\theta j} b_j$ denotes the total arrival rate into bundle $b$ (if offered) from customer type $\theta$.
- For $b \in B_\theta$, for any $\theta'$:

$$\overline{V}_{\theta' b} = \frac{\sum_j f_{\theta j} b_j V_{\theta' j}}{A_{\theta b}}$$

denotes the average value type $\theta'$ obtains from type $\theta$'s bundle $b$. Note that here we are assuming that the flow from bundle $b$ to the servers is consistent with the maximum flow $f$.

Note that given the maximum flow $\{f_{\theta j}\}$, the above are constants which we will use in our MILP formulation. We explain the decision variables and the constraints of the MILP briefly next:

**Decision Variable:**

-) $\{y_b,\ b \in \cup_\theta B_\theta\}$: These binary decision variables correspond to the possible service bundles for all customer types. A value of 1 indicates the bundle is offered and 0 indicates it is not offered.

-) $\{W_b,\ b \in \cup_\theta B_\theta\}$: These continuous non-negative decision variables correspond to the delay of bundles measure in units of (dis)utility up to a translation so that $\min_b W_b = 0$.

-) $\{U_\theta,\ \theta \in \Theta\}$: These continuous non-negative decision variables correspond to the utility of a type $\theta$ customer (up to a translation).

-) $\{W'_j,\ j \in [m]\}$: These decision variables correspond to the delay of server $j$ (and thus of all offered bundles containing server $j$), measured in units of (dis)utility and determined up to a translation.

OBJECTIVE:

$$\mathcal{W}^* := \min \sum_{\theta} \sum_{b \in B_\theta} A_{\theta,b} \cdot W_{\theta,b}$$

CONSTRAINTS:

**Feasibility of menu:**
$$\sum_{b \in B_\theta} b_j y_b = 1. \tag{3.8.1}$$

**Consistency of waiting times:**
$$W'_j - (1 - y_b)M \leq W_b \leq W'_j + (1 - y_b)M, \quad W_b \leq y_b M. \tag{3.8.2}$$

**Utility for each type:**
$$\overline{V}_{\theta,b} - W_b - (1 - y_b)M \leq U_\theta \leq \overline{V}_{\theta,b} - W_b + (1 - y_b)M. \tag{3.8.3}$$

**Incentive compatibility:**
$$U_{\theta'} \geq \overline{V}_{\theta',b} - W_b - (1 - y_b)M. \tag{3.8.4}$$

**Non-negativity of decision variables:**
$$\{W_b\}, \{W'_j\}, \geq 0 \quad \text{and} \quad \{y_b\} \in \{0, 1\}.$$

Figure 3.9: MILP for finding tailored menu with minimum average delay under maximum total value constraint.

Constraint (3.8.1) ensures that for each customer type $\theta$, a server $j$ with $f_{\theta j} > 0$ in the max value flow solution is offered in exactly one bundle intended for $\theta$. Constraint (3.8.2) ensures that (i) the delay disutility $W_b$ of any bundle $b$ that is offered and contains server $j$ equals the delay disutility $W'_j$ for server $j$ and (ii) the delay disutility for any bundle $b$ that is not offered is forced to 0 (so it contributes 0 to the objective). Constraint (3.8.3) ensures

that the utility of a customer type $\theta$ equals the utility of any offered bundle $b$ intended for $\theta$. Finally, (3.8.4) ensures that the utility of type $\theta'$ is at least the utility she derives from all offered bundles (which may or may not be intended for $\theta'$).

The objective minimizes the total delay disutility. The solution to the MILP will assign a set of service bundles to each customer type. Each assigned service bundle corresponds to a service class. Let $\{b_1, \ldots, b_s\}$ be the bundles selected by the MILP, so that without loss of generality we can assume that $0 = W_{b_1} \leq W_{b_2} \leq \cdots \leq W_{b_s}$. Making a similar assumption as used for Proposition 15, we can show the existence of a menu and an equilibrium such that at the heavy traffic limit bundle $b$ is served only by the servers in bundle $b$ and the limiting scaled mean delay of bundle $b$ is $\widehat{W}_b = \frac{W_b}{\delta} + \omega_0^*$. In other words, the objective of the MILP measures precisely the *additional delay disutility* compared to the minimum delay disutility experienced under a single CRP matching system.

The solution may potentially offer identical service bundles to multiple customer types. There would be no differences in terms of outcomes if the service provider were to offer distinct service classes for each customer type, or only offer each distinct service bundle once. The solution to the MILP may potentially assign multiple service bundles to a single customer type, in which case in equilibrium that customer type will be joining multiple service classes. Because of this, the Value Maximizing Tailored menu may have more distinct service classes than there are customer types.

Note that for every customer type $\theta$, we need to enumerate all $2^{S_\theta}$ bundles. Computationally this is not prohibitive if in the maximum value tree, the degree of each customer type is small. In Section 3.9 we present results from numerical experiments based on the MILP in Figure 3.9.

**3.8.2 Delay Minimizing Tailored Menus:** According to Proposition 4, any menu that induces a heavy traffic equilibrium with a single CRP component (such as the

Single Line menu) minimizes customers' limiting waiting times. In this section, we look at how to find a menu with higher matching values than the Single Line menu, while still maintaining a single CRP component at the heavy traffic equilibrium. We will narrow our focus to menus in which each customer type is joining a single service class in equilibrium in designing such a menu. This restriction means the arrival rates into the different service classes are fixed, which allows us to encode the flows between service classes and servers within a MILP. We will see in Section 3.9 that the menus we identify perform well even with this restriction.

Just like in the previous section, we will formulate this problem as a mixed-integer linear program. Consider a menu $M$ with $|\Theta| = n$ and let $\overline{V}_{\theta i}$ denote the average reward that a customer type $\theta$ is expected to receive by joining service class $i$. Since, $|\Theta| = n$, in what follows we will abuse notation and refer to service class $i \in [n]$ as the one targeted to customers of type $i \in \Theta$. Similarly, we will denote by $A_i$ the limiting arrival rate at class $i \in [n]$.

To ensure the incentive compatibility of the proposed menu, the service provider would like to design the menu in such a way that (i) it induces a single CRP component and (ii) the following IC condition is satisfied.

$$\overline{V}_{ii} \geq \overline{V}_{ik}, \quad \text{for all } k \in [n]. \tag{IC}$$

To formulate the service provider problem, we will rely on the quadratic program (**QP**) to approximate the steady-state matching probabilities for a given matching topology. Specifically, we propose the mixed integer linear program (MILP)[7] presented in Figure 3.10 to identify the matching probabilities that maximize the approximated average reward under the (IC) condition and the single CRP requirement.

7. This MILP is an extension of the one studied in Afèche et al. (2021).

OBJECTIVE:

$$\mathcal{V}^* := \max \sum_{ij} A_i \cdot p_{ij} \cdot V_{ij}$$

CONSTRAINTS:

**Approximate FCFS matching rates: KKT conditions of the (QP)**

$$\sum_{j\in[m]} p_{ij} = 1, \quad \sum_{i\in[n]} f_{ij} = \mu_j, \quad p_{ij} \le Z m_{ij}, \quad p_{ij} \le Z z_{ij}, \quad \nu_{ij} \le (n+m+1)Y \cdot (1 - z_{ij}), \tag{3.8.5a}$$

$$\mu_j(\theta_i + \gamma_j + \nu_{ij}) - Z(1 - m_{ij}) \le p_{ij} \le \mu_j(\theta_i + \gamma_j + \nu_{ij}) + Z(1 - m_{ij}), \tag{3.8.5b}$$

$$\text{where} \quad Y := \frac{1}{2}\max\left\{\frac{1}{A_{\min}}, \frac{1}{\mu_{\min}}\right\} \quad \text{and} \quad Z := A_{\max} \cdot \mu_{\max} \cdot \left(\frac{n}{A_{\min}} + \frac{m}{\mu_{\min}} + (n+m+1)^2 Y\right)$$

$$A_{\max} = \max_{i\in[n]}\{A_i\}, \quad A_{\min} = \min_{i\in[n]}\{A_i\}, \quad \mu_{\max} = \max_{j\in[m]}\{\mu_j\}, \quad \mu_{\min} = \min_{j\in[m]}\{\mu_j\}.$$

**Enforcing incentive compatibility condition** (IC): $$\sum_{j\in[m]} V_{ij}\,(p_{kj} - p_{ij}) \le 0. \tag{3.8.6}$$

**Enforcing a single CRP component: $n$ sets of constraints (indexed by $k \in \mathcal{C}$)**

$$\sum_{i\in\mathcal{C}} g_{ij}^{(k)} = \mu_j, \qquad \sum_{j\in\mathcal{S}} g_{ij}^{(k)} = A_i - \frac{\varepsilon}{n-1}, \quad g_{kj}^{(k)} = A_k + \varepsilon \qquad g_{ij}^{(k)} \le Z m_{i,j}, \tag{3.8.7}$$

$$\text{where } \varepsilon := \left(\prod_{i\in[n]} q_i \prod_{j\in[m]} q_j\right)^{-1} \quad \text{and } \tfrac{p_i}{q_i} = \tilde{A}_i,\ \tfrac{p_j}{q_j} = \mu_j \text{ are the rational number representations.}$$

**Non-negativity of decision variables:** $\{p_{ij}\}, \{\nu_{ij}\}, \{g_{ij}^{(k)}\} \ge 0$ and $\{m_{ij}\}, \{z_{ij}\} \in \{0,1\}$.

Figure 3.10: MILP for finding a tailored menu with maximum reward rate under minimum average delay constraint.

We explain the decision variables and the constraints of the MILP in brief next.

**Decision Variables:**

- $\{m_{ij},\ (i,j) \in [n] \times [m]\}$: These binary decision variables correspond to the matching topology.

- $\{p_{ij},\ (i,j) \in [n] \times [m]\}$ : These decision variables approximate the matching probabil-

ities on edge $(i,j)$ under the matching topology $\{m_{ij}\}$ and FCFS-ALIS matching.

- $\{\eta_i, (i \in [n]); \omega_j, (j \in [m]); \nu_{ij}, ((i,j) \in [n] \times [m])\}$ : These decision variables correspond to the dual variables for flow balance constraints (3.8.5a), and non-negativity constraint for $p_{ij}$, respectively, and are used to enforce the KKT conditions for the quadratic program (**QP**). Recall that the QP dictates that for some constants $\{\eta_i\}_{i\in[n]}$, $\{\gamma_j\}_{j\in[m]}$, and $\{\nu_{ij}\}_{(i,j)\in[n]\times[m]} \geq 0$, we have $p^*_{ij} = \mu_j(\theta_i + \gamma_j + \nu_{ij})$ and $f_{ij} \cdot \nu_{ij} = 0$ if $m_{ij} = 1$, and $f^*_{ij} = 0$ otherwise.

- $\{z_{ij},\ (i,j) \in [n] \times [m]\}$: The binary variable $z_{i,j}$ is used to enforce complementary slackness for the non-negativity constraint $p_{ij} \geq 0$: $z_{i,j} = 0$ enforces $p_{ij} = 0$ and $z_{i,j} = 1$ enforces $\nu_{i,j} = 0$.

- $\{g^{(k)}_{ij}, (i,j,k) \in [n] \times [m] \times [n]\}$ : These $n$ sets of flow variables (where the set is indexed by the superscript $k \in [n]$) are used to enforce the single CRP (equivalently minimum average delay) requirement. In words, the $k^{\text{th}}$ set of variables corresponds to the adjusted flows when we increase $A_k$ by a small $\varepsilon$, and reduce each $A_i$ for $i \neq k$ by $\frac{\varepsilon}{n-1}$.

Constraints (3.8.5a)-(3.8.5b) are the flow balance constraints. The constants $Y, Z$ ensure that the constraints impose the KKT conditions of (**QP**) for any matching topology $M$. These constraints and the non-negativity of $p_{ij}$ imply:

$$p_{ij} = \begin{cases} \mu_j(\theta_i + \gamma_j + \nu_{ij}), & m_{ij} = 1, \\ 0, & m_{ij} = 0. \end{cases}$$

Constraints (3.8.5a) and non-negativity of $p_{ij}$ and $\nu_{ij}$ imply the complementary slackness constraint $f_{ij} \cdot \nu_{ij} = 0$. Constraints (3.8.6) ensures the IC condition $\overline{V}_{ii} \geq \overline{V}_{ik}$ for all $i, k \in [n]$. Finally, the proof that the constraints (3.8.7) are necessary and sufficient to

ensure that the matching topology $M = [m_{ij}]$ induces a single CRP component can be found in Afèche et al. (2021).

## 3.9 Numerical Experiments

In Sections 3.7 and 3.8, we presented two classes of service menus that the service provider can use to design the service system, partition menus and tailored menus. In this section, we perform some preliminary numerical experiments to explore two questions. Firstly, we would like to understand for which preference structures the different approaches perform better or worse. Secondly, we would like to understand how what is the cost to the service provider of not being able to control customer strategies.

We begin by showing plots of the performance of Pareto efficient partition menus and tailored menus in the reward-delay quadrant for three particular instances of parameters. The customer valuations for servers were generated using the distribution $V_{\theta j} = \theta \times j + N(0, \sigma)$, where both $\theta$ and $j$ take values in $\{1, 2, 3, 4, 5\}$. (Recall that $\theta$ indexes customer types and $j$ servers.) Valuations are then translated so that $\min_{\theta j} V_{\theta j} = 0$, and scaled so that $\max_{\theta j} V_{\theta j} = 10$. We show results for $\sigma \in \{0, 2, 5\}$, $\Gamma = [1, 1, 1, 1, 1]$, $A = 5/18[2, 5, 1, 6, 4]$, and $a = 1/5[1, 1, 1, 1, 1]$. For comparison, we also show a bound on the performance of any menu using a linear programming relaxation of the problem that assumes that the service provider is able to decide the delays and matching rates for each customer type separately, and only the incentive compatibility constraints need to be satisfied. Details of the LP bound can be found in Appendix C. These plots provide some intuition about the relative performance of each menu. We will later show that this intuition applies quite generally, and does not depend on the particular valuations used to generate these plots. As we can see, the partition menus perform better relative to the tailored menus when there is less noise. The delay minimising tailored menu performs better as the noise increases. The value

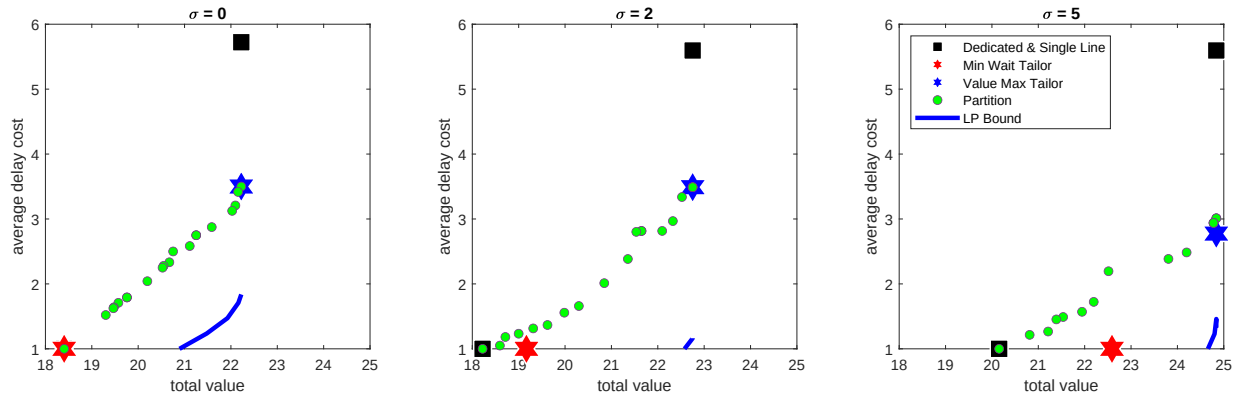



Figure 3.11: Performance of different menus when $V_{\theta j} = \theta \cdot j + N(0, \sigma)$ for $\sigma = 0, 2, 5$ in the average reward vs. average delay quadrant.

maximising tailored menu only performs better than some partition menus when noise is large.

Next, we compare the performance of the different mathematical programming approaches for different values of $\zeta$. Using the same valuation distributions and the same values of $\Gamma$, $A$, and $a$ as we used to generate Figure 3.11, we compare the performance of the chained partition menus and the tailored menus. For $\sigma = 2$ and $\sigma = 5$, we randomly generate 100 different instances of valuations, and report the average performance across all the instances. For $\sigma = 0$, since there is no randomness, we only have one instance, and so we report the performance of that instance directly. For each value of $\zeta$ and each valuation instance, we find the optimal partition menu. For each instance, the tailored menus do not change depending on $\zeta$, only the objective function values do.

For each menu, we report the ratio

$$\frac{\widehat{V}^* - \zeta \widehat{W}^*}{\widehat{V}_{\text{VM}} - \zeta \widehat{W}_{\text{DM}}}.$$

That is, we are comparing the performance of each menu with that of the first best outcome,

in which matching values are maximized and delays are minimized simultaneously. In the last row of the table, we report the same ratio for the LP bound.

| $\sigma$ | 0 | | | | 2 | | | |
|---|---|---|---|---|---|---|---|---|
| $\zeta$ | 0 | 0.05 | 0.25 | 0.5 | 0 | 0.05 | 0.25 | 0.5 |
| CP | 1.000 | 0.972 | 0.864 | 0.806 | 1.000 | 0.975 | 0.896 | 0.848 |
| VM | 1.000 | 0.972 | 0.851 | 0.683 | 1.000 | 0.975 | 0.867 | 0.720 |
| DM | 0.828 | 0.826 | 0.818 | 0.806 | 0.909 | 0.908 | 0.904 | 0.898 |
| LP | 1.000 | 0.979 | 0.897 | 0.806 | 1.000 | 0.988 | 0.955 | 0.931 |

| $\sigma$ | 5 | | | |
|---|---|---|---|---|
| $\zeta$ | 0 | 0.05 | 0.25 | 0.5 |
| CP | 1.000 | 0.980 | 0.919 | 0.878 |
| VM | 1.000 | 0.981 | 0.899 | 0.789 |
| DM | 0.957 | 0.957 | 0.955 | 0.953 |
| LP | 1.000 | 0.994 | 0.979 | 0.971 |

Table 3.2: Average performance of different menus when $V_{\theta j} = \theta \cdot j + N(0, \sigma)$ for $\sigma = 0, 2, 5$ relative to the LP bound.

These results show that the intuitions from Figure 3.11 hold true across many instances, as well as providing some new intuitions. For low values of $\zeta$, the optimal partition menu performs at least as well as the tailored menus, and when there is no noise, the partition menu performs at least as well as the tailored menus for all values of $\zeta$. For high values of $\zeta$, the delay minimizing tailored menu performs at least as well as the partition menus and the value maximising tailored menus, regardless of how much noise there is. The performance of the delay minimising tailored menu also improves relative to the LP bound as the noise increases. The value maximising tailored menu only outperforms the partition menus when noise is large, and $\zeta$ is small.

By comparing the performance of each menu with the first best outcome, we are able to see how much the service provider is losing out by not being able to control the customer strategies as well as the menu. When the service provider can control the customer strategies

as well as the menu, they are always able to achieve an outcome arbitrarily close to the first best outcome, by offering all of the service classes in the Dedicated menu, and an additional service class served by all servers. The strategies that will achieve close to a first best outcome would be to have customers generally use their value maximising strategy, while diverting a small amount of flow to the service class served by all servers.

We can see the performance of all menus relative to the first best outcome decreases with $\zeta$. This would continue for all values of $\zeta$ for the value maximising menu, as the value maximising menu is a fixed menu with delays worse than the first best delays. The delay minimising menu achieves the minimum possible delay, so for large enough values of $\zeta$ we would see the performance of the delay minimising menu begin to improve relative to the first best outcome, as the delays become more and more dominant in the objective. We would expect the same thing to happen for the LP bound and the optimal partition menu, as for large enough $\zeta$, the LP bound and the optimal partition menu would have minimum possible delays, which can always be achieved in an incentive compatible way by offering a single line menu. We can also see that the performance of all menus relative to the first best outcome improves as noise increases.

## 3.A Chapter 3 Proofs

**3.A.1 Proof of Theorem 3:** We will use Kakutani's Fixed Point Theorem to show that a $\Delta$-equilibrium exists for $\Delta = 0$ when $|\alpha| < |\mu|$.

**Theorem 9** (Kakutani's Fixed Point Theorem)**.** *Let $Q$ be a non-empty, compact and convex subset of some Euclidean space $\mathbb{R}^n$. Let $F : Q \to 2^Q$ be a set-valued funtcion on $X$ with the following properties:*

- *$F$ has a closed graph;*
- *$F(q)$ is non-empty and convex for all $q \in Q$.*

*Then $F$ has a fixed point.*

We will apply Kakutani's Fixed Point Theorem to a best response function $F : Q \to 2^Q$, which we will now construct. In constructing the best response function $F$, it will be useful to extend our definitions of $W_i(q)$, $p_{ij}(q)$, and $U_{\theta i}(q)$ to strategy profiles $q$ for which the system does not admit a steady state under a FCFS-ALIS service discipline.

To do this, we introduce the concept of a reduced service system. Fix a strategy profile $q$, and let $\lambda_I(q)$ denote the arrival rate into the set $I$ of service classes under $q$. The strategy profile $q$ need not admit a steady state distribution. Thus, we define $\overline{\mathscr{I}} \subseteq [n]$ as the minimal set of unstable service classes and let $\overline{\mathscr{J}} = S(\overline{\mathscr{I}})$ be the servers compatible with $\overline{\mathscr{I}}$ under menu $M$. That is, $\overline{\mathscr{I}}$ is the minimal set satisfying $\forall \mathscr{I}' \cap \overline{\mathscr{I}} = \emptyset$:

$$\lambda_{\mathscr{I}'}(q) < \mu_{S(\mathscr{I}') \cap \mathscr{J}},$$

where $\mathscr{J} := [m] \setminus \overline{\mathscr{J}}$, and $\mathscr{I} := [n] \setminus \overline{\mathscr{I}}$. The set $\overline{\mathscr{I}}$ is unique, and a non-constructive method of identification is as follows: Let $f^*$ denote the optimal value of the maximum

flow in the network with nodes $[m] \cup [n]$, maximum inflow into service node $i$ of $\lambda_i(q)$ and maximum capacity of server node $j$ of $\mu_j$. If for a service class $i$, there exists some $\epsilon_i > 0$, such that the new maximum flow obtained by increasing the inflow into service class $i$ by $\epsilon_i$ is $f^* + \epsilon_i$ then $i \in \mathscr{I}$, otherwise $i \in \overline{\mathscr{I}}$.

The reduced service system is given by only keeping the service classes $\mathscr{I}$ and servers $\mathscr{J}$. The menu $M^{\mathscr{I}}$ is the submatrix of $M$ with rows corresponding to service classes in $\mathscr{I}$, and columns corresponding to $\mathscr{J}$. We use $\lambda^{\mathscr{I}}(q)$ to denote the vector of arrival rates for service classes $\mathscr{I}$, and $\Gamma^{\mathscr{J}}$ as the service rate vector for the servers in $\mathscr{J}$. Note that the sets $\mathscr{I}$ and $\mathscr{J}$ are a function of the strategy profile $q$. We will denote them by $\mathscr{I}(q)$ and $\mathscr{J}(q)$ when this dependence is not clear from the context.

We now use this reduced system to define $p_{ij}(q)$ and $W_i(q)$ for arbitrary strategy profiles $q$ (potentially for which the system is unstable), and the best response map. By definition, the reduced service system $(\lambda^{\mathscr{I}}(q), \Gamma^{\mathscr{J}}, M^{\mathscr{I}})$ is stable, and hence admits steady state mean waiting times which we denote by $W_i^{\mathscr{I}}(q)$ for $i \in \mathscr{I}$, and matching probabilities, defined to be $p_{ij}^{\mathscr{I}}(q)$ for $i \in \mathscr{I}, j \in \mathscr{J}$. For $i \in \mathscr{I}, j \in \mathscr{J}$, we set $p_{ij}(q) = p_{ij}^{\mathscr{I}}(q)$. For all other combinations of $(i,j) \in [n] \times [m]$, we set $p_{ij}(q) = 0$. Similarly for $i \in \mathscr{I}$ we set $W_i(q) = W_i^{\mathscr{I}}(q)$, and for all $i \notin \mathscr{I}$ we set $W_i(q) = \infty$.

With these extended definitions of $p_{ij}(q)$ and $W_i(q)$, we can also extend the definition of $U_{\theta i}(q)$ which allows us to define the best response set of each customer type for any strategy profile $q$. Let $B_\theta(q)$ be the set of all service classes which maximize the utility of customers in class $\theta$ given strategy profile $q$, that is,

$$B_\theta(q) = \left\{ i \in [n] \;\middle|\; i \in \operatorname*{argmax}_{i'} U_{\theta i'}(q) \right\}.$$

For any customer type $\theta$ and any strategy profile $q$, let

$$F_\theta(q) = \text{conv}(\{e_i | i \in B_\theta(q)\}).$$

We then define the best response function $F(q)$ as

$$F(q) = \underset{\theta \in \Theta}{\times} F_\theta(q). \tag{3.A1}$$

It is clear from the definition of $F$ that $F(q)$ is non-empty and convex for all $q$. All that remains to be shown in order to use Kakutani's Fixed Point Theorem is that the graph of $F$ is closed. To do this, we will show that the graph of $F$ contains all of its limit points.

Let $\{q_k\}_{k\in\mathbb{N}}$ and $\{q_k^*\}_{k\in\mathbb{N}}$ be a sequences of strategy profiles such that $q_k \to q$ and $q_k^* \to q^*$, where $q$ and $q^*$ are strategy profiles, and $q_k^* \in F(q_k)$ for all $k \in \mathbb{N}$. To show that the graph of $F$ contains all of its limits points, we need to show that $q^* \in F(q)$. To do this, we need to show that for all $\theta \in \Theta$ and $i \in [n]$ such that $q_{\theta i}^* > 0$, $q_{\theta i}^* \in B_\theta(q)$.

Consider any pair $(\theta, i)$ such that $q_{\theta i}^* > 0$. Then there must exist some $K \in \mathbb{N}$ such that for all $k > K$, $q_{k\theta i}^* > 0$. This implies that $i \in B_\theta(q_k)$ for all $k > K$, or, $U_{\theta i}(q_k) \geq U_{\theta i'}(q_k)$ for all $i' \in [n]$. To show that $U_{\theta i}(q) \geq U_{\theta i'}(q)$ for all $i' \in [n]$, it suffices to show that $U_{\theta i}(q_k) \to U_{\theta i}(q)$ for all $\theta, i$ as $k \to \infty$.

Let $\mathscr{I}(q)$ denote the set of stable service classes under limiting strategy profile $q$. Since $q_k \to q$ implies $\lambda_I(q_k) \to \lambda_I(q)$ for all subsets $I \subseteq [n]$, it is true that $\mathscr{I}(q) = \liminf_{k\to\infty} \mathscr{I}(q_k)$. Further, for all $i \notin \mathscr{I}(q)$ (the unstable service classes), $W_i(q_k) \to \infty$ and hence $\lim_{k\to\infty} U_{\theta i}(q_k)$ $= -\infty = U_{\theta i}(q)$ for $i \notin \mathscr{I}$. For the remaining classes, $i \in \mathscr{I}(q)$, there exists some $K$, such that for all $k \geq K$, $i \in \mathscr{I}(q_k)$. Thus by continuity of the steady-state distribution for FCFS-ALIS model (for stable matching topologies), for $i \in \mathscr{I}(q)$, $p_{ij}(q_k) \to p_{ij}(q)$ and $W_i(q_k) \to W_i(q)$, and hence $U_{\theta i}(q_k) \to U_{\theta i}(q)$.

This completes the proof that the graph of $F$ is closed. So Kakutani's Fixed Point Theorem applies, and we know that there exists some strategy profile $q$ satisfying $q \in F(q)$. □

**3.A.2 Proof of Proposition 13:** To compute the average scaled waiting time $\overline{W}_{\max}$ under the Dedicated menu we need to impose the equilibrium conditions. First, we need to ensure that the limiting arrival rate to class $i$ converges (from below) to $\mu_i$ for $i = 1, 2$. Thus, under the assumption $A_1 > \mu_1$, we must have some customer type $\bar{\theta} \in \Theta_1$ that is randomizing between joining the dedicated queue for server 1 and the dedicated queue for server 2. In equilibrium, this randomization strategy should be such that this customer type is indeed indifferent between joining these two service classes. To identify the type $\bar{\theta}$ we need to rank the customers' types in $\Theta_1$ according to the value of $\Delta V_\theta$. For this, let $K_1 = |\Theta_1|$ denote the cardinality of $|\Theta_1|$ and let us index its elements $\Theta_1 = \{\theta_1, \theta_2, \ldots, \theta_{K_1}\}$ in such a way that $\Delta V_{\theta_1} \geq \Delta V_{\theta_2} \geq \cdots \geq \Delta V_{\theta_{K_1}}$. In case of ties, i.e., if $\Delta V_{\theta_i} = \Delta V_{\theta_{i+1}}$, then we require $V_{\theta_i 1} \geq V_{\theta_{i+1} 1}$.

Let us denote by $\bar{\kappa}$ the index that defines $\bar{\theta}$, that is, $\bar{\theta} = \theta_{\bar{\kappa}}$. Now, if type-$\theta_{\bar{\kappa}}$ customers are indifferent between the two service classes then we must have that customers' type $\theta_k$ (with $k < \bar{\kappa}$) prefer class 1 over class 2. Hence, to ensure that the arrival rate to class $i$ converges to $\mu_i$ from below the value of $\bar{\kappa}$ must be equal to

$$\bar{\kappa} := \min \left\{ \kappa \in [K_1] \colon \sum_{k=1}^{\kappa} A_{\theta_k} > \mu(1) \right\}$$

and a fraction

$$\hat{q}_{\bar{\kappa}} := \frac{\mu_1 - \sum_{k=1}^{\bar{\kappa}-1} A_{\theta_k}}{\sum_{k=1}^{\bar{\kappa}} A_{\theta_k}}$$

of the type-$\theta_{\bar{\kappa}}$ customers must select class 1.

Also, a type-$\theta_{\bar{\kappa}}$ is indifferent between the two dedicated queues if $\Delta V_{\theta_{\bar{\kappa}}} = \delta\,(\widehat{W}_1^{\mathrm{D}} - \widehat{W}_2^{\mathrm{D}})$,

where $\widehat{W}_i^{\mathrm{D}}$ is the scaled steady-state mean waiting time of class $i = 1, 2$ under the Dedicated menu. Furthermore, from Theorem 1, we know that $\widehat{W}_i^{\mathrm{D}} = 1/\tilde{\gamma}_i$, where $\tilde{\gamma}_i$ is the scaled capacity slack of class $i$. But the sum of the slacks of the two classes is equal to the aggregated system lack, that is, $\tilde{\gamma}_1 + \tilde{\gamma}_2 = |a|$. Using this identity, the indifference condition $\Delta V_{\theta_{\bar{\kappa}}} = \delta\,(\widehat{W}_1^{\mathrm{D}} - \widehat{W}_2^{\mathrm{D}})$ leads to the following equation on $\tilde{\gamma}_1$:

$$\Delta V_{\theta_{\bar{\kappa}}} = \delta\,\left(\frac{1}{\tilde{\gamma}_1} - \frac{1}{|a| - \tilde{\gamma}_1}\right).$$

Solving for $\tilde{\gamma}_1$ and plugging back the solution in the values for $\widehat{W}_1^{\mathrm{D}}$ and $\widehat{W}_2^{\mathrm{D}}$ we get

$$\widehat{W}_1^{\mathrm{D}} = \frac{2\,\Delta V_{\theta_{\bar{\kappa}}}}{2\,\delta + |a|\,\Delta V_{\theta_{\bar{\kappa}}} - \sqrt{4\,\delta^2 + \left(|a|\,\Delta V_{\theta_{\bar{\kappa}}}\right)^2}}$$

$$\widehat{W}_2^{\mathrm{D}} = \frac{2\,\Delta V_{\theta_{\bar{\kappa}}}}{|a|\,\Delta V_{\theta_{\bar{\kappa}}} - 2\,\delta + \sqrt{4\,\delta^2 + \left(|a|\,\Delta V_{\theta_{\bar{\kappa}}}\right)^2}}.$$

Finally, to obtain the value of $\overline{W}_{\max}$ we note that in a Dedicated menu a flow of $\mu_i$ customers join class $i$ in the heavy traffic limit, $i = 1, 2$. It follows that

$$\overline{W}_{\max} = \left(\frac{\mu_1}{\mu_1 + \mu_2}\right) \widehat{W}_1^{\mathrm{D}} + \left(\frac{\mu_2}{\mu_1 + \mu_2}\right) \widehat{W}_2^{\mathrm{D}}. \tag{3.A2}$$

Let us turn to the derivation of $\overline{W}_{\mathrm{med}}$, the average customers' delay performance achieved by the Full and $N_1$ menus. We can compute this value using a similar line arguments as the one we just used to compute $\overline{W}_{\max}$. Again the equilibrium conditions imply that customers type-$\theta_{\bar{\kappa}}$ (same as above) must randomize between joining class 1 or class 3 and the randomization probability must equal $\hat{q}_{\theta_{\bar{\kappa}}}$ to ensure that the arrival rate to class $i$ converges to $\mu_i$ from below in the heavy traffic limit for $i = 1, 3$. The main difference with the Dedicated menu is than under the $N_1$ menu the two CRP components are not longer disconnected but

rather chained. Thus, Theorem 1 implies that the scaled waiting time of class 3 is equal to $\widehat{W}_3^{\mathrm{N}_1} = 1/|a|$. In addition, a type-$\theta_{\bar{\kappa}}$ is indifferent between the two classes if $\Delta V_{\theta_{\bar{\kappa}}} = \delta\,(\widehat{W}_1^{\mathrm{N}_1} - \widehat{W}_3^{\mathrm{N}_1})$ and so $\widehat{W}_1^{\mathrm{N}_1} = 1/|a| + \Delta V_{\theta_{\bar{\kappa}}}/\delta$. Combining these values with the fact that a flow of $\mu_1$ customers join class 1 in equilibrium we get that

$$\overline{W}_{\,\mathrm{med}} = \frac{1}{|a|} + \left(\frac{\mu_1}{\mu_1+\mu_2}\right)\frac{\Delta V_{\theta_{\bar{\kappa}}}}{\delta}. \tag{3.A3}$$

□

**3.A.3 Proof of Proposition 11:** Since $\hat{q}^*$ is a heavy traffic equilibrium, there exists a direction $\hat{\phi}^* \in \mathbb{R}^{|\Theta|}$ satisfying the conditions in Definition 12. For $\epsilon > 0$, let us define the strategy $q^{(\epsilon)} = \hat{q}^* + \hat{\phi}^*\,\epsilon$. To prove the result, we will show that $q^{(\epsilon)}$ satisfies condition (a) in the proposition for an appropriate sequence $(\Delta_\epsilon)_{\epsilon>0}$ that converges to 0 as $\epsilon \downarrow 0$. Specifically, we need to show that for all $\theta \in \Theta$ and for all $i, k \in [n]$

$$q_{\theta i}^{(\epsilon)}\left(U_{\theta i}\big(W^{(\epsilon)(\epsilon)}(q^{(\epsilon)}), p^{(\epsilon)}(q^{(\epsilon)})\big) - U_{\theta k}\big(W^{(\epsilon)(\epsilon)}(q^{(\epsilon)}), p^{(\epsilon)}(q^{(\epsilon)})\big)\right) \geq -\Delta^{(\epsilon)}. \tag{3.A4}$$

Since $\hat{q}^*$ is a heavy traffic equilibrium converging along the direction $\hat{\phi}^*$, conditions (a) and (b) in Definition 12 imply that the left-hand side of this inequality converges to a non-negative limit as $\epsilon \downarrow 0$. It follows then that for all $\Delta > 0$ there exists an $\varepsilon(\Delta) > 0$ such that for all $\epsilon \in (0, \varepsilon(\Delta))$ we have

$$q_{\theta i}^{(\epsilon)}\left(U_{\theta i}\big(\widehat{W}^{(\epsilon)}(q^{(\epsilon)}), p^{(\epsilon)}(q^{(\epsilon)})\big) - U_{\theta k}\big(\widehat{W}^{(\epsilon)}(q^{(\epsilon)}), p^{(\epsilon)}(q^{(\epsilon)})\big)\right) \geq -\Delta.$$

Furthermore, we can always select the mapping $\varepsilon(\Delta) > 0$ to be continuous and monotonically increasing in a neighborhood $(0, \bar{\Delta})$, for some $\bar{\Delta} > 0$, and such that $\lim_{\Delta\downarrow 0}\varepsilon(\Delta) = 0$. Then, for $\epsilon$ small enough we can define $\Delta^{(\epsilon)} := \varepsilon^{-1}(\epsilon/2)$. It follows that $\lim_{\epsilon\downarrow 0}\Delta^{(\epsilon)} = 0$ and that the inequality in (3.A4) is satisfied. □

**3.A.4 Proof of Proposition 12:** Let us use a slight abuse of notation and denote by $\widehat{W}^*_k$ the limiting scaled waiting times of all service classes that belong to $\mathbb{C}_k$ for $k \in [K]$. Define the vector $\widehat{W}'$ such that $\widehat{W}'_k = \widehat{W}^*_k - \widehat{W}^*_1 + 1/|a|$. We next show that $(\hat{q}^*, \widehat{W}', \hat{p}^*)$ is a heavy traffic equilibrium that weakly Pareto dominates $(\hat{q}^*, \widehat{W}^*, \hat{p}^*)$ since $\widehat{W}' \leq \widehat{W}^*$. To this end, we note that by Proposition 3 $\widehat{W}'$ is implementable by a chained DAG on $\mathbb{C}$. Let $\widetilde{\gamma}' = (\widetilde{\gamma}'_1, \ldots, \widetilde{\gamma}'_K)$ be the vector of cumulative capacity slacks that implement $\widehat{W}'$(see Definition 6) and define the direction of convergence to heavy traffic $\phi'$ as a solution to system of linear equations $\widetilde{\gamma}' = a\,\hat{q}^* - A\,\phi'$. By this construction, one case see that $(\hat{q}^*, \widehat{W}', \hat{p}^*)$ satisfies conditions (a) and (b) in Definition 12. Indeed, (a) holds since the sequence of pre-limit strategy profiles $q'^{(\epsilon)} = \hat{q}^* + \epsilon\,\phi'$ satisfies $\widehat{W}' = \lim_{\epsilon \downarrow 0} \widehat{W}^{(\epsilon)}(q'^{(\epsilon)})$ and $\hat{p}^* = \lim_{\epsilon \downarrow 0} p^{(\epsilon)}(q'^{(\epsilon)})$. On the other hand, (b) holds trivially since $\widehat{W}'_k$ is a translation of $\widehat{W}^*$. $\square$

**3.A.5 Proof of Theorem 4:** First, under a Single Line menu every customer – irrespective of its type– is served by server $j$ with probability $\mu_j/|\mu|$. It follows that,

$$\overline{V}^{\text{SL}} = \sum_{\theta \in \Theta} \frac{A_\theta}{|A|} \sum_{j \in [m]} \frac{\mu_j}{|\mu|}\, V_{\theta j}.$$

On the other hand, let $\widehat{W}^* = (\widehat{W}^*_i)_{i \in [n]}$ and $\hat{p}^* = [\hat{p}^*_{ij}]_{i \in [n], j \in [m]}$ be the limiting steady-state waiting times and matching probabilities under the pair $(M, \hat{q}^*)$. It follows that

$$\overline{V}(M, \hat{q}^*) = \sum_{\theta \in \Theta} \frac{A_\theta}{|A|} \sum_{i \in [n]} \hat{q}^*_{\theta i} \sum_{j \in [m]} \hat{p}^*_{ij}\, V_{\theta j}.$$

Now, let $\hat{\lambda}^*_i$ be the equilibrium arrival rate to class $i \in [n]$ under $(M, \hat{q}^*)$, that is,

$$\hat{\lambda}^*_i = \sum_{\theta \in \Theta} A_\theta\, \hat{q}^*_{\theta i}.$$

and let us define the strategy $q = [q_{\theta i}]_{\theta\in\Theta, i\in[n]}$ by

$$q_{\theta i} = \frac{\hat{\lambda}_i^*}{|\mu|}.$$

Note that $q$ is feasible strategy (i.e., $q \in \mathcal{Q}$) since $|\hat{\lambda}^*| = |A| = |\mu|$.

By the equilibrium condition that $\hat{q}^*$ satisfies, there is no customer type $\theta$ that would strictly prefer to use strategy $(q_{\theta i})_{i\in[n]}$ instead of $(\hat{q}^*_{\theta i})_{i\in[n]}$. It follows that

$$\begin{aligned}
\sum_{i\in[n]} \hat{q}^*_{\theta i}\Big(\sum_{j\in[m]} \hat{p}^*_{ij}\, V_{\theta j} - \delta\, \widehat{W}_i\Big) &\geq \sum_{i\in[n]} q_{\theta i}\Big(\sum_{j\in[m]} \hat{p}^*_{ij}\, V_{\theta j} - \delta\, \widehat{W}_i\Big) \qquad \text{(equilibrium condition)} \\
&= \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|A|}\Big(\sum_{j\in[m]} \hat{p}^*_{ij}\, V_{\theta j} - \delta\, \widehat{W}_i\Big) \qquad \text{(definition of } \hat{\lambda}_i^*) \\
&= \sum_{j\in[m]} \frac{V_{\theta j}}{|\mu|} \sum_{i\in[n]} \hat{\lambda}_i^*\, \hat{p}^*_{ij} - \delta \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|\mu|}\, \widehat{W}_i \\
&= \sum_{j\in[m]} \frac{\mu_j}{|\mu|}\, V_{\theta j} - \delta \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|\mu|}\, \widehat{W}_i. \qquad \text{(since } \textstyle\sum_{i\in[n]} \hat{\lambda}_i^*\, \hat{p}^*_{ij} = \mu_j)
\end{aligned}$$

Let multiply both sides of the inequality by $A_\theta/|A|$ and sum over $\theta \in \Theta$ to get

$$\overline{V}(M, \hat{q}^*) - \delta \sum_{\theta\in\Theta} \frac{A_\theta}{|A|} \sum_{i\in[n]} \hat{q}^*_{\theta i}\, \widehat{W}_i \geq \overline{V}^{\text{SL}} - \delta \sum_{\theta\in\Theta} \frac{A_\theta}{|A|} \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|\mu|}\, \widehat{W}_i.$$

But

$$\begin{aligned}
\sum_{\theta\in\Theta} \frac{A_\theta}{|A|} \sum_{i\in[n]} \hat{q}^*_{\theta i}\, \widehat{W}_i &= \sum_{i\in[n]} \frac{\widehat{W}_i}{|A|} \sum_{\theta\in\Theta} A_\theta\, \hat{q}^*_{\theta i} \\
&= \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|\mu|}\, \widehat{W}_i = \sum_{\theta\in\Theta} \frac{A_\theta}{|A|} \sum_{i\in[n]} \frac{\hat{\lambda}_i^*}{|\mu|}\, \widehat{W}_i \quad \text{(recall that } |A| = |\mu|)
\end{aligned}$$

and so we conclude that $\overline{V}(M, \hat{q}^*) \geq \overline{V}^{\text{SL}}$. $\square$

**3.A.6 Proof of Theorem 5:** We will now use approximate complementary slackness to show that any equilibrium set of flows under the partition menu gives an approximately optimal solution to the max flow problem (**Max-flow**), and that the optimality gap goes to 0 as $\epsilon \downarrow 0$.

Recall that $f_{\theta j}^{(\epsilon)}$ is the flow of customers type $\theta$ served by server $j$ in the max flow formulation (**Max-flow**). Similarly, $\eta_\theta^{(\epsilon)}$ and $\omega_j^{(\epsilon)}$ are the dual variables for the flow balance constraint for customer type $\theta$ and the capacity constraint of server $j$, respectively, in the dual problem (3.6.2). For a feasible primal solution $f_{\theta j}^{(\epsilon)}$ and a feasible dual solution $(\eta_\theta^{(\epsilon)}, \omega_j^{(\epsilon)})$ to satisfy approximate complementary slackness, it is sufficient that

$$\begin{aligned} 0 \quad &\leq \left(\mu_j - \textstyle\sum_\theta f_{\theta j}^{(\epsilon)}\right) \omega_j^{(\epsilon)} \quad \leq \varepsilon_j \\ 0 \quad &\leq \left(\eta_\theta^{(\epsilon)} + \omega_j^{(\epsilon)} - V_{\theta j}\right) f_{\theta j}^{(\epsilon)} \quad \leq \varepsilon_{\theta j} \end{aligned} \tag{3.A5}$$

for all $\theta$ and $j$, and $\varepsilon_j, \varepsilon_{\theta j} \ll 1$. With these we can show approximate optimality of $f_{\theta j}^{(\epsilon)}$, namely, show that $\sum_{\theta,j} V_{\theta j}\, f_{\theta j}^{(\epsilon)} \approx \bar{\mathcal{V}}^{(\epsilon)}$. Formally, weak duality gives:

$$\sum_{\theta,j} f_{\theta j}^{(\epsilon)}\, V_{\theta j} \leq \bar{\mathcal{V}}^{(\epsilon)} \leq \sum_\theta \alpha_\theta^{(\epsilon)}\, \eta_\theta^{(\epsilon)} + \sum_j \mu_j\, \omega_j^{(\epsilon)}.$$

Weak complementary slackness and primal/dual feasibility imply:

$$\begin{aligned} \sum_\theta \alpha_\theta^{(\epsilon)}\, \eta_\theta^{(\epsilon)} + \sum_j \mu_j\, \omega_j^{(\epsilon)} &\leq \sum_{\theta,j} f_{\theta j}^{(\epsilon)} \eta_\theta^{(\epsilon)} + \sum_j \left(\omega_j^{(\epsilon)} \sum_\theta f_{\theta j}^{(\epsilon)} + \varepsilon_j\right) \\ &= \sum_{\theta,j} f_{\theta j}^{(\epsilon)} \left(\eta_\theta^{(\epsilon)} + \omega_j^{(\epsilon)}\right) + \sum_j \varepsilon_j \\ &\leq \sum_{\theta,j} f_{\theta j}^{(\epsilon)}\, V_{\theta j} + \sum_{\theta,j} \varepsilon_{\theta j} + \sum_j \varepsilon_j. \end{aligned}$$

Combining, we get

$$\bar{\mathcal{V}}^{(\epsilon)} - \left( \sum_{\theta,j} \varepsilon_{\theta j} + \sum_{j} \varepsilon_j \right) \leq \sum_{\theta,j} f_{\theta j}^{(\epsilon)} V_{\theta j} \leq \bar{\mathcal{V}}^{(\epsilon)}. \tag{3.A6}$$

Let us now show that for any equilibrium arrival rates ${f_{\theta j}}^{(\epsilon)}$[1] for the $\epsilon^{\text{th}}$ system, we can construct a dual solution such that approximate complementary slackness holds. To this end, let $\widehat{W}_j^{(\epsilon)}$ be the equilibrium limited scaled waiting time for the service class served by server $j$. For all $j \in [m]$ and $\theta \in \Theta$, we let

$$\omega_j{}^{(\epsilon)} = \delta\, \widehat{W}_j^{(\epsilon)} \qquad \text{and} \qquad \eta_\theta{}^{(\epsilon)} = \max_j \left\{ V_{\theta j} - \omega_j{}^{(\epsilon)} \right\} \tag{3.A7}$$

denote a feasible dual solution. We know that under any equilibrium, $f_{\theta k}^{(\epsilon)} > 0$ only ifj

$$j \in \arg\max_{j'} \left\{ \overline{V}_{\theta j} - \delta\, \widehat{W}_j^{(\epsilon)} \right\}, \quad \text{that is,} \quad j \in \arg\max_{j'} \left\{ \overline{V}_{\theta j} - \omega_j^{(\epsilon)} \right\}.$$

Therefore $f_{\theta j}^{(\epsilon)} > 0$ only if $\eta_\theta{}^{(\epsilon)} + \omega_j{}^{(\epsilon)} - V_{\theta j} = 0$. So *exact* complementary slackness holds for the first set of dual constraints: $\varepsilon_{\theta j} = 0$ for all $\theta, j$. Furthermore, for the primal constraints, we use the fact that under the Dedicated menu service class $j$ operates as a single M/M/1 queue with arrival rate $\sum_\theta {f_{\theta j}}^{(\epsilon)}$ and service capacity $j$. It follows that the (non-scaled) waiting time in service class $j$ equals $W_j^{(\epsilon)} = 1/(\mu_j - \sum_\theta {f_{\theta j}}^{(\epsilon)})$. Thus, since the scaled waiting time in service class $j$ satisfies $\widehat{W}_j^{(\epsilon)} = \epsilon\, W_j^{(\epsilon)}$, which implies

$$0 \leq \left( \mu_j - \sum_\theta {f_{\theta j}}^{(\epsilon)} \right) \omega_j{}^{(\epsilon)} = \delta\, \epsilon. \tag{3.A8}$$

1. We know that an equilibrium exists from Theorem 3.

Or, approximate complementary slackness holds with $\varepsilon_j = \delta\,\epsilon$. Then (3.A6) implies

$$\bar{\mathcal{V}}^{(\epsilon)} - \sum_{\theta,j} f_{\theta j}{}^{(\epsilon)} V_{\theta j} \leq \delta\,\epsilon\,m.$$

So as $\epsilon \downarrow 0$, the difference between the value $V^{(\epsilon)} := \sum_{\theta,j} f_{\theta j}{}^{(\epsilon)} V_{\theta j}$ achieved in equilibrium under the Dedicated menu and the upper bound $\bar{\mathcal{V}}^{(\epsilon)}$ converges to zero. □

**3.A.7 Proof of Theorem 6:** To get the main idea across, we first assume that for all customer types $\theta$, the rewards $V_{\theta j}$ are distinct. Later in the proof we remove this assumption.

Let the service classes be labelled so that service class $j$ is the dedicated service class for server $j$. We begin by claiming that in any heavy traffic equilibrium, with limiting probabilities $\hat{p}^*$, for any service class $i$ there can be at most one server $j$ with $\hat{p}^*_{ij} > 0$. Suppose not, and assume that there are $\hat{p}^*_{ij} > 0$ and $\hat{p}^*_{ij'} > 0$ (for simplicity assume there are only two such servers, the proof generalizes easily). Then servers $j$ and $j'$ must be in the same CRP component. Further service classes dedicated to servers $j$ and $j'$ must also be in the same CRP component and hence the limiting scaled mean delay of service classes $j$ and $j'$ equal the limiting scaled mean waiting time for service class $i$. It can happen that the arrival rate into the dedicated service classes $j$ or $j'$ could be zero, however we can still talk about the virtual waiting time of a customer joining these service classes, and the statement would hold for the limiting scaled virtual waiting time.

Now by assumption, at least one of the dedicated service classes $j$ or $j'$ give strictly higher matching value and no higher delay to any customer type joining class $i$, and therefore this can not be an equilibrium.

The equilibrium matching system therefore looks as follows: the service classes are partitioned into $m+1$ sets $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_m$, such that the classes in $\mathcal{C}_0$ have asymptotically negligible demand: $\mathcal{C}_0 = \{i \in [n] | \Lambda_i = 0\}$. For $j \geq 1$, the service classes in $\mathcal{C}_j$ have asymp-

totically non-negligible flow to only server $j$: $\hat{p}^*_{ij} = 1$ for $i \in \mathcal{C}_j$. Again, for this outcome to be an equilibrium, we must have the limiting scaled mean waiting for all service classes within each $\mathcal{C}_j$ ($j \geq 1$) to be equal, and therefore in heavy-traffic limit any customer type will be indifferent between any service class within $\mathcal{C}_j$. We will denote by $\widehat{W}^*_j$ the limiting scaled mean waiting time for service classes in $\mathcal{C}_j$ for $j \geq 2$. To summarize, for any service class $i \in \mathcal{C}_j$, the limiting utility obtained by a customer type $\theta$ is $U_{\theta i} = V_{\theta j} - \delta\widehat{W}_j$ .

Define:

$$f_{\theta j} = \sum_{i \in \mathcal{C}_j} A_\theta \hat{q}^*_{\theta i}. \tag{3.A9}$$

as the total flow from customer type $\theta$ to server $j$ (through service classes in $\mathcal{C}_j$). Denoting the utility of type $\theta$ as

$$U_\theta = \max_i U_{\theta i} = \max_j V_{\theta j} - \delta\widehat{W}_j,$$

best response condition gives $f_{\theta j} > 0$ only if $j \in \text{argmax}_{j'} \left\{ V_{\theta j'} - \delta\widehat{W}_{j'} \right\}$ or equivalently,

$$f_{\theta j} \cdot \left( U_\theta - V_{\theta j} + \delta\widehat{W}_j \right) = 0.$$

But these are precisely the complementary slackness conditions for the maximum value flow linear program (**Max-flow**) in the proof of Theorem 5. Since $[f_{\theta j}]$ is a feasible primal solution, the complementary slackness conditions imply that it is also an optimal, and hence value maximizing, flow.

**Removing assumption on rewards:** To summarize what we have done so far, we showed that we can use the equilibrium to define the flow matrix $[f_{\theta j}]$, customer utilities $U_\theta$, and server delays $\delta\widehat{W}_j$ so that they are feasible primal dual solutions to the maximum value flow problem (**Max-flow**) and satisfy complementary slackness conditions. We now

show we can do so without the restriction on rewards.

Fix a heavy-traffic equilibrium $\hat{q}^*$, and the resulting limiting probabilities $\hat{p}^*$. Define $\widehat{W}_i$ as the limiting mean scaled waiting time of the CRP component that service class $j$ belongs to. Recall the definitions:

$$U_{\theta i} = \sum_j \hat{p}^*_{ij} V_{\theta j} - \delta \widehat{W}_i,$$

and by the best response condition, the customer utility is defined by

$$U_\theta = \max_i U_{\theta i}.$$

Define $\mathcal{S}^*(i) = \{j : \hat{p}^*_{ij} > 0\}$ as the "effective"set of servers for service class $i$. Let customer type $\theta$ join a service class $i$ (that is, $\hat{q}^*_{\theta i} > 0$) so that $U_\theta = U_{\theta i}$. Suppose $|\mathcal{S}^*(i)| \geq 2$ (the case $|\mathcal{S}^*(i) = 1|$ is vacuously true for the argument). Then we must have that for $j, j' \in \mathcal{S}^*(i)$, $\widehat{W}_j = \widehat{W}_{j'}$ since $j, j'$ are in the same CRP component. It must also be the case that for all $j \in \mathcal{S}^*(i)$, $V_{\theta j}$ are equal. If not, then type $\theta$ can deviate to the dedicated service class for the server $j \in \mathcal{S}^*(i)$ with highest reward – this strictly improves the reward and does not incur any further delay disutility. Therefore, for all $j \in \mathcal{S}^*(i)$

$$V_{\theta j} - \delta \widehat{W}_j = U_{\theta i} = U_\theta.$$

Define the total flow from customer type $\theta$ to server $j$, $f_{\theta j}$, as:

$$f_{\theta j} = A_\theta \sum_i \hat{q}^*_{\theta i} \hat{p}^*_{ij}.$$

The preceding arguments imply $f_{\theta j} > 0$ only if $U_\theta = V_{\theta j} - \delta \widehat{W}_j$. We thus again find that $[f_{\theta j}]$, $U_\theta$, $\widehat{W}_j$ define feasible primal-dual solution to (**Max-flow**) satisfying complementary slackness, and hence the flow $[f_{\theta j}]$ maximizes the matching reward. $\square$

**3.A.8 Proof of Theorem 7:** Suppose the service provider is able to achieve a first best outcome by offering the menu $M*$. As there may be multiple equilibria, throughout this proof we will use equilibrium to mean the first best outcome achieving equilibrium. Let $q^*_{\theta i}$ be the equilibrium strategies, and let $p^*_{ij}$ be the equilibrium matching rates. We will also let $f^*\theta j$ be the equilibrium flows between customer types and servers.

Since a first best outcome is achieved, we know that $f^*_{ij}$ constitute an optimal solution to **Max-flow** with $\epsilon = 0$. We will begin by showing that the positive flows from this solution form a connected graph.

Since a first best outcome is achieved, we know that in equilibirum there is a single CRP component, and hence there is a connected graph between service classes and servers. Since every customer type is joining at least one service class, this implies that there is a path between every customer type and every server, which in turn implies that there is a path between every pair of customer types. Hence the flows between customer types and servers also form a connected graph.

Next we will show that no customer type prefers the **Max-flow** matching outcome of any other customer type. Take any two customer types $\hat{\theta}$ and $\tilde{\theta}$. We will show that $\hat{\theta}$ does not prefer that matching outcome of $\tilde{\theta}$. We will let $\hat{V}$ be the matching value that $\hat{\theta}$ achieves from their equilibrium strategy. Since there is a single CRP component, we know that $\hat{V}$ is the value that $\hat{\theta}$ gains from every service class they are joining in equilibrium, and that $\hat{V}$ is at least as large as the value $\hat{\theta}$ would achieve from joining any other service class.

The value $\hat{\theta}$ gains from $\tilde{\theta}$'s **Max-flow** matching outcome is

$$\begin{aligned} V(\hat{\theta}, \tilde{\theta}) &= \sum_i q^*_{\tilde{\theta} i} \sum_j p_{ij} V_{\hat{\theta},j} \\ &\leq \sum_i q^*_{\tilde{\theta} i} \hat{V} \\ &= \hat{V}. \end{aligned}$$

Thus $\hat{\theta}$ prefers their own matching outcome to that of any other customer type. This completes the proof. □

**3.A.9 Proof of Corollary 3:** The proof of this corollary follows the proof of Theorem 5 for the Dedicated menu essentially verbatim by reinterpreting an individual server in the Dedicated menu by a super-server for each of the partitions with a service capacity equals to the sum of the capacities of the servers in the partition. The only small difference in the proof relates to equation (3.A8). Specifically, since super-server $k$ does not operates exactly as an M/M/1, it is not longer true that the (non-scaled) waiting time $W_k^{(\epsilon)}$ for service class $\mathcal{C}_k$ is equal to $1/(\mu_{\mathcal{S}_k} - \sum_\theta \hat{f}_{\theta k})$. However, we next show that for all $\epsilon \leq \min_j\{\mu_j\}/((m+1)\,|a|)$, we have

$$\left(\mu_{\mathcal{S}_k} - \sum_\theta \hat{f}_{\theta k}\right) W_k^{(\epsilon)} \leq 2,$$

which suffices to complete the rest of the steps in the proof of Theorem 5.

To this end, we use the fact that service class $\mathcal{C}_k$ is a single-line multi-server queue with arrival rate $\sum_\theta f^{(\epsilon)}_{\theta k}$ and system utilization $\rho_k^{(\epsilon)} := \sum_\theta f^{(\epsilon)}_{\theta k}/\mu_{\mathcal{S}_k}$. Let us denote by $m_k$ the number of servers in $\mathcal{S}_k$ and by $\{\pi_k^{(\epsilon)}(s)\}$ the stationary distribution of the number of customers in service class $k$ (including those in service). The average number of customers

in this class satisfies

$$L_k^{(\epsilon)} = \sum_{s=0}^{\infty} s\,\pi_k^{(\epsilon)}(s) \le \sum_{s=0}^{m_k-1} m_k\,\pi_k^{(\epsilon)}(s) + \sum_{s=m_k}^{\infty} s\,\pi_k^{(\epsilon)}(s) = m_k + \sum_{s=m_k}^{\infty} (s-m_k)\,\pi_k^{(\epsilon)}(s)$$
$$= m_k + \sum_{s=0}^{\infty} s\,\pi_k^{(\epsilon)}(m_k+s) = m_k + \sum_{s=0}^{\infty} s\,(\rho_k^{(\epsilon)})^s\,\pi_k^{(\epsilon)}(m_k) = m_k + \frac{\rho_k^{(\epsilon)}}{(1-\rho_k^{(\epsilon)})^2}\,\pi_k^{(\epsilon)}(m_k).$$

In the second-to-last equality we have used the brith-death property structure of the system, which implies $\pi_k(s) = (\rho_k^{(\epsilon)})^{s-m_k}\,\pi_k^{(\epsilon)}(m_k)$ for all $s \ge m_k$. We also use this fact to get an upper bound on the value of $\pi_k^{(\epsilon)}(m_k)$ as follows:

$$1 = \sum_{s=0}^{\infty} \pi_k^{(\epsilon)}(s) \ge \sum_{s=m_k}^{\infty} \pi_k^{(\epsilon)}(s) = \sum_{s=m_k}^{\infty} (\rho_k^{(\epsilon)})^{s-m_k}\,\pi_k^{(\epsilon)}(m_k) = \frac{\pi_k^{(\epsilon)}(m_k)}{1-\rho_k^{(\epsilon)}} \quad \Longrightarrow \quad \pi_k^{(\epsilon)}(m_k) \le 1-\rho_k^{(\epsilon)}.$$

Combining this inequality, the inequality for $L_k^{(\epsilon)}$ above and the fact that $W_k^{(\epsilon)} = L_k^{(\epsilon)} / \sum_\theta f_{\theta k}^{(\epsilon)}$ (by Little's law) we get

$$\Big(\mu_{\mathcal{S}_k} - \sum_\theta f_{\theta k}^{(\epsilon)}\Big)\, W_k^{(\epsilon)} \le \Big(\mu_{\mathcal{S}_k} - \sum_\theta f_{\theta k}^{(\epsilon)}\Big)\, \frac{m_k}{\sum_\theta \hat{f}_{\theta k}} + 1.$$

By stability we must have $\sum_\theta {f_{\theta k}}^{(\epsilon)} \ge |\alpha^{(\epsilon)}| - (|\mu| - \mu_{\mathcal{S}_k}) = \mu_{\mathcal{S}_k} - |a|\,\epsilon$. We use this inequality to upper bound the right-hand side above to get

$$\Big(\mu_{\mathcal{S}_k} - \sum_\theta f_{\theta k}^{(\epsilon)}\Big)\, W_k^{(\epsilon)} \le \frac{|a|\,m_k\,\epsilon}{\mu_{\mathcal{S}_k} - |a|\,\epsilon} + 1.$$

Finally, it is not hard to check that for $\epsilon \le \min_j\{\mu_j\}/((m+1)\,|a|)$ the upper bound above is less than or equal to 2. □

**3.A.10 Proof of Proposition 15:** Let $(\hat{q}^*, \widehat{W}^{\mathrm{PB}}, \hat{p}^*)$ be the heavy traffic equilibrium under the pure partition menu. From Proposition 14 and the assumption $\omega_1 < \omega_2$ we have

that $1/|a| < \widehat{W}_1^{\text{PB}} < \widehat{W}_2^{\text{PB}}$. Thus, $(\hat{q}^*, \widehat{W}^{\text{PB}}, \hat{p}^*)$ satisfies the conditions in Proposition 12. It follows that we can construct another heavy traffic equilibrium $(\hat{q}^*, \widehat{W}^{\text{CB}}, \hat{p}^*)$ that (weakly) Pareto dominates $(\hat{q}^*, \widehat{W}^{\text{PB}}, \hat{p}^*)$ by chaining the CRP components in the pure partition menu. Furthermore, from the proof of Proposition 12 we have that $\widehat{W}_1^{\text{CB}} = 1/|a|$ and $\widehat{W}_k^{\text{CB}} = \widehat{W}_k^{\text{PB}} - \widehat{W}_1^{\text{PB}} + 1/|a|$ as required. Finally, it follows trivially that the two heavy traffic equilibria $(\hat{q}^*, \widehat{W}^{\text{PB}}, \hat{p}^*)$ and $(\hat{q}^*, \widehat{W}^{\text{CB}}, \hat{p}^*)$ produce the same matching value $\bar{\mathcal{V}}$ since they have the same limiting strategy profile $\hat{q}^*$ and matching probabilities $\hat{p}^*$. $\square$

## 3.B Upper Bound LP

Here we include the LP used to find an upper bound under FCFS-ALIS scheduling on the performance of any menu used in section Section 3.9.

The following are the decision variables used in the LP. formulation:

-) $p_{\theta j}$: probability that customer type $\theta$ is served by server $j$.

-) $f_{\theta j}$: flow of type-$\theta$ customers to server $j$.

-) $W_\theta$: waiting time for type $\theta$ customers.

OBJECTIVE:

$$\sum_{\theta\in\Theta} A_\theta \sum_{j\in[m]} p_{\theta j} V_{\theta j} - \zeta \sum_{\theta\in\Theta} A_\theta W_\theta \tag{3.A1}$$

CONSTRAINTS:

**Flow balance:**
$$\sum_j p_{\theta j} = 1, \qquad \sum_\theta A_\theta p_{\theta j} = \mu, \qquad f_{\theta j} = A_\theta p_{\theta j}. \tag{3.A2}$$

**Waiting time constraint:**
$$W_\theta \geq \frac{1}{\sum_\theta a_\theta} \tag{3.A3}$$

**Incentive compatibility:**
$$\sum_j \left(p_{\theta j} - p_{\theta' j}\right) V_{\theta j} + W_{\theta'} - W_\theta \geq 0. \tag{3.A4}$$

**Non-negativity of decision variables:**
$$\{p_{\theta j}\}, \{f_{\theta j}\}\{W_\theta\} \geq 0. \tag{3.A5}$$

Figure 12: LP for finding an upperbound on the performance of any menu.

# CHAPTER 4

# CONCLUDING REMARKS AND FUTURE DIRECTIONS

In Chapter 2, we studied the performance of multi-class multi-server bipartite queueing systems under a FCFS-ALIS service discipline by extending the heavy traffic analysis introduced in Afèche et al. (2021) for a similar class of systems. In Theorem 1 we have provided a general characterization of the mean steady-state waiting time for each service class. Our characterization relies on decomposing the queueing system into a collection of complete resource pooling (CRP) components and identifying the connectivity among these CRP components in the form of a directed acyclic graph (DAG). Interestingly, only the knowledge of this DAG together with the capacity slack in each CRP component is enough to derive the mean steady-state waiting time for all service classes. We have also studied the steady-state matching probabilities between service classes and servers and showed in Theorem 2 that only the limiting values of arrival and service rates influence these matching probabilities. This is in direct contrast to the behaviour of the mean steady-state waiting times, which are also affected by the direction of convergence to heavy traffic. To illustrate this point, we have provided a numerical example that shows that small changes to the arrival rates in a heavily congested system can have large impacts on the average delays. We use our results regarding steady-state outcomes to explore some questions regarding the design of queueing systems. In doing this, we find that when service providers are looking to minimise expected delays and have complete control over the design of the menu, then they should implement a menu that induces a single CRP component.

In Chapter 3, we have taken the first steps towards studying the design of service systems with congestion in the presence of strategic customers. A key message of our results is that more is not always better – restricting customer choice is as important as offering richer service classes. On the constructive side, we presented a mathematical programming

approach to menu design. Our experimental results demonstrate that menus with one service class per type are sufficient to find good menus. In particular, there exist menus which achieve minimum average delay, and at the same time achieve matching values quite close to the optimal. Such menus are appealing for two reasons (i) their simplicity, and (ii) the ability to search within this space through a mathematical programming approach.

Our work points towards several promising research directions. An area that deserves further investigation is the relationship between delays and the underlying matching topology in our bipartite queueing system when arrival rates into the different service classes are fixed. In Section 2.5.3, we demonstrate that adding more connectivity to the system can lead to a deterioration in the average waiting time of customers, exhibiting a form of Braess's paradox, despite neither customers nor servers acting strategically. Mathematically, this negative effect happens when adding an additional arc to the menu increases the probability of a topological order with higher conditional delays. Theorem 1 characterizes waiting time delays and can be used to identify an optimal flexibility structure as a combinatorial optimization problem over the collection of directed acyclic graphs (DAGs) associated with a particular set of CRP components.

Several challenging problems remain towards building a full theory of service menu design with strategic customers; we mention a few. First, we saw empirical evidence that menus with one service class per customer type can perform well, but lack theoretical bounds. Second, we need a better characterization of the effect of reward structure on the trade-off between matching value and delay on the Pareto frontier. An even simpler question is the following: Given a reward matrix, what is the minimum loss in matching value necessary under a single CRP constraint? This question is quite similar in spirit to the notion of *price of envy-freeness* in the literature on envy-free cake cutting. Again, our experiments indicate this to be small, but it is possible to construct extreme examples where the matching value under a single CRP constraint can be an arbitrarily small fraction of the optimal which makes

our experimental results even more intriguing. A third question is on the non-uniqueness of equilibrium. We avoided equilibrium selection problems via the notion of provider-preferred equilibrium, but menus with unique equilibria may offer practical advantages such as robustness. Alternatively, one could explore the pessimistic formulation further. While the matching value outcomes Dedicated and Pure Partition Menus are the same for all equilibria, giving us some understanding of their performance even in the pessimistic case, both classes of Tailored menus we have developed do not have natural extensions for the pessimistic formulation. An interesting direction to explore would be how to approach the menu design question when taking a pessimistic approach. Fourth, our results rely on the quasilinear structure of the utility function and homogeneous delay costs. With heterogeneous delay costs, the value optimality of the dedicated menu also breaks down. Extension to more general reward structures, or better yet, menus which are robust to the misspecification of utility functions is also an important and challenging direction. Finally, the vast literature on the design of price/lead-time menus relies on the *achievable region method* queueing systems where the service provider has full flexibility to dynamically route customers. A similar tool for FCFS-ALIS queueing systems could further expand the menu design settings to which we can apply a mathematical programming approach.

Considering both questions of menu design and performance analysis, there are alternative modelling choices that could be worth exploring. For example, while we have focused on conventional heavy-traffic scaling in this work, a many-server scaling may be more appropriate for certain application settings, such as public housing and healthcare, where many identical servers are available. Furthermore, we have primarily examined steady-state outcomes, but in real-world scenarios, conditions often change frequently, making it unclear if a steady-state will be achieved. Therefore, studying the transient behaviour of bipartite queueing systems could also be of interest.

# BIBLIOGRAPHY


I. Adan and G. Weiss. Exact FCFS matching rates for two infinite multitytpe sequences. *Operations Research*, 60(2):475–489, 2012. 8, 9

I. Adan and G. Weiss. A skill based parallel service system under FCFS-ALIS – steady state, overloads and abandonments. *Stochastic Systems*, 4(1):250–299, 2014. 8, 9, 16, 17, 26, 85, 86, 94

I. Adan, A. Bušić, J. Mairesse, and G. Weiss. Reversibility and further properties of FCFS infinite bipartite matching. *Mathematics of Operations Research*, 43(2):598–621, 2018a. 8

I. Adan, I. Kleiner, R. Righter, and G. Weiss. FCFS parallel service systems and matching models. *Performance Evaluation*, 127-128:253–272, 2018b. 8

I. Adan, M. Boon, and G. Weiss. Design heuristic for parallel many server systems. *European Journal of Operations Research*, 273(1):259–277, 2019. 8

P. Afèche. Incentive-compatible revenue management in queueing systems: Optimal strategic delay. *Manufacturing & Service Operations Manamgement*, 15(3):423–443, 2013. 11

P. Afèche and J. Pavlin. Optimal price/lead-time menus for queues with customer choice: Segmentation, pooling, and strategic delay. *Manamgement Science*, 62(8):2412–2436, 2016. 11

P. Afèche, R. Caldentey, and V. Gupta. On the optimal design of a bipartite matching queueing system. *Operations Research*, 2021. ix, 2, 3, 7, 8, 9, 10, 11, 19, 20, 21, 22, 23, 25, 26, 27, 32, 36, 40, 43, 45, 47, 54, 59, 64, 66, 67, 68, 94, 95, 128, 131, 153

M. Akan, O. Alagoz, B. Ata, F. Erenay, and A. Said. A broader view of the liver allocation system incorporating disease evolution. *Operations Research*, 60(4):757–770, 2012. 12

M. Akbarpour, S. Li, and S. O. Gharan. Thickness and information in dynamic matching markets. 2018. 12

R. Anderson, I. Ashlagi, D. Gamarnik, and Y. Kanoria. Efficient dynamic barter exchange. *Operations Research*, 65(6):1446–1459, 2017. 12

N. Arnosti and P. Shi. Design of lotteries and waitlists for affordable housing allocation. 2018. 12

N. Arnosti, R. Johari, and Y. Kanoria. Managing congestion in matching markets. 2018. 12

I. Ashlagi, M. Burq, P. Jaillet, and V. Manshadi. On matching and thickness in heterogeneous dynamic markets. 2018b. 12

I. Ashlagi, F. Monachou, and A. Nikzad. Optimal dynamic allocation: Simplicity through information design. https://ssrn.com/abstract=3610386, 2021. 11

I. Ashlagi, J. Leshno, P. Qian, and A. Saberi. Price discovery in waiting lists: A connection to stochastic gradient descent. Technical report, University of Chicago, 2022. 11

R. Atar. Scheduling control for queueing systems with many servers: Asymptotic optimality in heavy traffic. *The Annals of Applied Probability*, 15(4):2606–2650, 2005. 11

R. Atar. A diffusion regime with nondegenerate slowdown. *Operations Research*, 60(2): 490–500, 2012. 6

M. Baccara, A. Collard-Wexler, L. Felli, and L. Yariv. Child-adoption matching: Preferences for gender and race. *American Economic Journal: Applied Economics*, 6(6):133–158, 2014. 12

M. Baccara, S. Lee, and L. Yariv. Optimal dynamic matching. 2018. 12

A. Bassamboo, R. Randhawa, and J. V. Mieghem. A little flexibility is all you need: On the asymptotic value of flexible capacity in parallel queuing systems. *Operations Research*, 60 (6):1423–1435, 2012. 12

S. Bell and R. Williams. Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Electron. J. Probab.*, 10:1044–1115, 2005. 11

S. L. Bell and R. J. Williams. Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. *The Annals of Applied Probability*, 11(3):608–649, 2001. 6

D. Bertsimas, V. Farias, and N. Trichakis. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, 61(1):73–87, 2013. 12

F. Bloch and D. Cantala. Dynamic assignment of objects to queuing agents. *American Economic Journal: Microeconomics*, 9(1):88–122, 2017. 12

A. Bušić, V. Gupta, and J. Mairesse. Stability of the bipartite matching model. *Advances in Applied Probability*, 45(2):351–378, 2013. 8

R. Caldentey and E. Kaplan. A heavy traffic approximation for queues with restricted customer-service matchings. Unpublished manuscript, 2002. 7, 8

R. Caldentey, E. Kaplan, and G. Weiss. FCFS infinite bipartite matching of severs and customers. *Adavances on Applied Probability*, 41(3):695–730, 2009. 8, 94

R. Caldentey, V. Gupta, and L. A. Hillas. Designing service menus for bipartite queueing systems. 2023. 7

C. Comte. Dynamic load balancing with tokens. *Computer Communications*, 144:76–88, 2019. 11

J. Dai and T. Tezcan. Optimal control of parallel server systems with many servers in heavy traffic. *Queueing Systems*, 59:95–134, 2005. 11

Y. Ding, T. McCormick, and M. Nagarajan. A fluid model for an overloaded bipartite queueing system with heterogeneous matching utility. 2018. 12

E. Duffin. Foster care in the u.s. - number of children waiting for adoption 2007-2021. *Statista*, 2022. 1

M. Fazel-Zarandi and E. Kaplan. Approximating the first-come, first-served stochastic matching model with ohm's law. *Operations Research*, 6:1423–1432, 2018. 7, 8, 95

K. Gardner and R. Righter. Product forms for fcfs queueing models with arbitrary server-job compatibilities: An overview. *Queueing Systems*, 96:3–51, 2020. 8, 9

L. Green. A queueing system with general-use and limited-use servers. *Operations Research*, 33(1):168–185, 1985. 8

I. Gurvich and A. Ward. On the dynamic control of matching queues. *Stochastic Systems*, 4(2):479–523, 2014. 12

I. Gurvich and W. Whitt. Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing Service Oper. Management*, 11(2):237–253, 2009. 11

I. Gurvich and W. Whitt. Service-level differentiation in many-server service systems via queue-ratio routing. *Operations Research*, 58(2):316–328, 2010. 11

S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations research*, 29(3):567–588, 1981. 6

J. Harrison. Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete review policies. *The Annals of Applied Probability*, 8(3):822–848, 1998. 11

J. M. Harrison. Brownian models of queueing networks with heterogeneous customer populations. In *Stochastic differential systems, stochastic control theory and applications*, pages 147–186. Springer, 1988. 6

J. M. Harrison and M. J. Lopez. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems Theory Appl.*, 33:339–368, 1999. 6, 7, 11

L. A. Hillas, R. Caldentey, and V. Gupta. Heavy traffic analysis of multi-class bipartite queueing systems under fcfs. Technical report, University of Chicago, 2023. 2, 4

D. A. Hurtado Lange and S. T. Maguluri. Heavy-traffic analysis of queueing systems with no complete resource pooling. *Mathematics of Operations Research*, 47(4):3129–3155, 2022. 7

W. C. Jordan and S. C. Graves. Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594, 1995. 12

E. Kaplan. Managing demand for publich housing. 1984. ORC Technical Report # 183, MIT. 8

E. Kaplan. A public housing queue with reneging and task-specific servers. *Decision science*, 19:383–391, 1988. 8

J. F. Kingman. On queues in heavy traffic. *Journal of the Royal Statistical Society: Series B (Methodological)*, 24(2):383–392, 1962. 6

H. J. Kushner and Y. Chen. Optimal control of assignment of jobs to processors under heavy traffic. *Stochastics: An International Journal of Probability and Stochastic Processes*, 68 (3-4):177–228, 2000. 7

J. Leshno. Dynamic matching in overloaded waiting lists. 2017. 12

C. Maglaras and A. Zeevi. Pricing and design of differentiated services: Approximate analysis and structural insights. *Operations Research*, 53(2):242–262, 2005. 11

J. Mairesse and P. Moyal. Stability of the stochastic matching model. *Journal of Applied Probability*, 53:1064–1077, 2017. 8

A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized c$\mu$-rule. *Operations Research*, 52(6):836–855, 2004a. 6

A. Mandelbaum and S. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized c$\mu$ rule. *Operations Research*, 52(6):836–855, 2004b. 11

R. Marler and J. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, Apr. 2004. 123

P. Moyal and O. Perry. On the instability of matching queues. *The Annals of Applied Probability*, 27(6):3385–3434, 2017. 8

M. Nazari and A. Stolyar. Optimal control of general dynamic matching systems. 2016. 12

H. Nazerzadeh and R. Randhawa. Near-optimality of coarse service grades for customer differentiation in queueing systems. *Production and Operations Management*, 27(23):578–595, 2018. 11

V. Pesic and R. Williams. Dynamic scheduling for parallel server systems in heavy traffic: Graphical structure, decoupled workload matrix and some sufficient conditions for solvability of the brownian control problem. *Stochastic Systems*, 6(1):26–89, 2016. 7

E. Plambeck. Optimal leadtime differentiation via diffusion approximations. *Operations Research*, 52(2):213–228, 2004. 11

R. Rogerson, R. Shimer, and R. Wright. Search-theoretic models of the labor market: A survey. *Journal of Economic Literature*, 43(4):959–988, 2005. 12

B. Schwartz. Queueing models with lane selection: A new class of problems. *Operations Research*, 22(2):331–339, 2004. 8

V. Shah and G. de Veciana. Asymptotic independence of servers' activity in queueing systems with limited resource pooling. *Queueing Systems*, 83(1-2):13–28, 2016. 7

J. Sheridan. She sought an affordable housing voucher in 1993. this chicago alderman just reached the top of the waitlist. *Chicago Tribune*, 2022. 1

C. Shi, Y.Wei, and Y. Zhong. Process flexibility for multi-period production systems. *Operations Research (forthcoming)*, 2018. 12

V. Slaugh, M. Akan, O. Kesten, and M. U. Ünver. The pennsylvania adoption exchange improves its matching process. *Interfaces*, 46(2):133–158, 2016. 12

R. Talreja and W. Whitt. Fluid models for overloaded multi-class many-service queueing systems with fcfs routing. *Management Science*, 54(1):1513–1527, 2008. 7

J. N. Tsitsiklis and K. Xu. On the power of (even a little) resource pooling. *Stochastic Systems*, 2(1):1–66, 2012. 12

J. N. Tsitsiklis and K. Xu. Flexible queueing architectures. *Operations Research*, 65(5): 1398–1413, 2017. 12

M. Unver. Dynamic kidney exchange. *Rev. Econom. Stud.*, 77(1):372–414, 2010. 12

J. Van Mieghem. Price and service discrimination in queuing systems: Incentive compatibility of gc$\mu$ scheduling. *Management Science*, 46(9):1249–1267, 2000. 11

S. M. Varma and S. T. Maguluri. Transportation polytope and its applications in parallel server systems. https://arxiv.org/abs/2108.13167, 2021. 8

R. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing Service Oper. Management*, 7(4):276–294, 2005. 11

A. Ward and M. Armony. Blind fair routing in large-scale service systems with heterogeneous customers and servers. *Operations Research*, 61(1):228–243, 2013. 11

W. Whitt. Heavy traffic limit theorems for queues: a survey. In *Mathematical Methods in Queueing Theory*, pages 307–350. Springer, 1974. 6

R. J. Williams. On dynamic scheduling of a parallel server system with complete resource pooling. *Fields Institute Communications*, 28(49-71):5–1, 2000. 6

S. Zenios, G. Chertow, and L. Wein. Dynamic allocation of kidneys to candidates on the transplant waiting list. *Operations Research*, 48(4):549–569, 2000. 12