Queueing Systems (2024) 106:239–284
https://doi.org/10.1007/s11134-024-09903-4

Heavy traffic analysis of multi-class bipartite queueing systems under FCFS

Lisa Aoki Hillas[1] · **René Caldentey**[2] · **Varun Gupta**[2]

Received: 30 March 2023 / Revised: 20 January 2024 / Accepted: 23 January 2024 /
Published online: 2 March 2024

Abstract

This paper examines the performance of multi-class, multi-server bipartite queueing systems, where each arriving customer is compatible with only a subset of servers. We focus on the system's performance under a first-come, first-served-assign longest idle server service discipline. In this discipline, an idle server is matched with the compatible customer who has been waiting the longest, and a customer who can be served by multiple idle servers is routed to the server that has been idle for the longest period. We analyse the system under conventional heavy-traffic conditions, where the traffic intensity approaches one from below. Building upon the formulation and results of Afèche et al. (Oper Res 70(1):363–401, 2022), 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 customer classes and demonstrate that a much wider range of waiting time outcomes is achievable. Furthermore, we establish that the matching probabilities, i.e. the probabilities of different customer 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 customer classes and servers, finding that a service provider who has complete control over the matching can design a delay-minimizing matching by considering only the limiting arrival rates. When some constraints on the compatibility structure exist, the direction of convergence to heavy-traffic affects which compatibility structure minimizes delay. Additionally, we discover that the bipartite matching queueing system exhibits a form of Braess's paradox, where adding more

✉ Lisa Aoki Hillas
lisa.hillas@auckland.ac.nz

René Caldentey
rene.caldentey@chicagobooth.edu

Varun Gupta
varun.gupta@northwestern.edu

[1] ISOM Department, The University of Auckland, Auckland, New Zealand

[2] Computer Science Department, Northwestern University Evanston, Evanston, USA

connectivity to an existing system can lead to higher average waiting times, despite the fact that neither customers nor servers act strategically.

Keywords Multi-class queueing system · First-come-first-served · Bipartite matching · Steady-state analysis

1 Introduction

In this paper, we analyse the performance of multi-class bipartite queuing systems under an FCFS-ALIS service discipline.[1] Multi-class bipartite queueing systems are ubiquitous in modelling a variety of operations, such as call centres, healthcare, manufacturing, and public housing, among others. However, these models can be both analytically and computationally intractable, making questions of performance analysis and system design difficult to answer. Heavy traffic scaling can be used to provide approximations of these systems that are much simpler to analyse and reveal fundamental properties of the system.

The specific model we consider has n different queues, each served by a distinct subset of the m available servers. We will refer to a queue as a "*customer class*", and the collection of all customer classes will constitute a "*service menu*" or simply a "*menu*". 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 customer class. Each customer class in the menu has a particular set of servers they can be served by. Each server may potentially be compatible with multiple customer classes. Servers serve the customer classes they are compatible with according to a FCFS discipline. 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. If a customer arrives to a customer class and multiple servers they are compatible with are idle, then the server that has been idle the longest will be assigned to serve them.

The FCFS service discipline is simple to implement and is widely used in practice. It is particularly appealing in applications where fairness is a concern, such as healthcare delivery and public housing allocations. Similarly, in applications in which servers correspond to workers, fairness may motivate the choice of ALIS as a server assignment policy. Alternatively, when servers correspond to physical resources such as public housing units, it might be preferred to avoid leaving any one unit empty and unused for long periods of time. This again motivates the use of an ALIS server assignment policy.

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

[1] The acronym FCFS-ALIS stands for "first-come, first-served-assign longest idle server". This means that when a server becomes idle it selects the customer who has been waiting the longest among those it can serve. Similarly, a customer that can be served by multiple idle servers selects the server that has been idle the longest.

This paper extends the work of Afèche et al. [3], who study a similar model to ours. Their formulation uses a specific heavy traffic scaling, which limits the range of outcomes that the model can produce. In particular, Afèche et al. [3] restrict themselves to asymptotic limits in which the direction of convergence to heavy traffic keeps constant the proportion of customers arriving into the different queues (similar heavy traffic limits are discussed in [15, 21]). In contrast, our paper considers more general heavy traffic limits, allowing for accurate approximations of a broader range of scenarios and providing new insights into an efficient management of multi-class multi-server queueing systems. For instance, in Sect. 5, we demonstrate that minor variations in the direction of convergence to the heavy traffic limit can significantly impact customers' waiting times in the pre-limit. Additionally, we show that the limiting matching probabilities depend solely on the limiting arrival rates, and are insensitive to the specific direction of convergence to heavy traffic. Our perturbation analysis sheds new light on how system managers can reduce waiting time delays by inducing marginal changes in the arrival rates of various customer classes, without compromising the quality of the matching between customer classes and servers.

We also extend the results in Afèche et al. [3] by allowing some customer classes to have no arrivals at the heavy traffic limit. Our primary motivation for considering this generalization is to study systems with strategic customers, i.e. customers who can choose their class type upon arrival based on waiting time delays and matching probabilities. In such scenarios, it is possible that in equilibrium, customers completely avoid joining some of the available customer classes. Similarly, a service provider might nudge arriving customers to join certain specific customer classes by creating others that are unattractive. Interestingly, we will demonstrate that, despite having zero limiting arrival rates, these vanishing customer classes can significantly impact the waiting delays of other classes.

Finally, we also explore some questions regarding the design of the compatibility between customer classes and servers. 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 minimizing compatibility structure. When there are some constraints on the compatibility structure, then the particular approach to heavy traffic does affect which compatibility structure minimizes delay.

Related Literature Heavy traffic approximations have long been used to simplify the study of intractable queueing systems. Early works in this area include [19, 30]. 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., [13] or [4]. Motivated by mathematical tractability as well as by the fact that many real-world service systems operate under high levels of congestion,[2] we will study the performance of our multi-class multi-server bipartite queuing system operating under conventional heavy traffic conditions. In positioning

[2] For example, the Chicago Housing Authority reported more than 170,000 families waiting for public housing in 2021. Similarly, in the same year, about 113,589 children in the United States were waiting to

our work within the extensive heavy traffic literature, it is worth noting that our analysis exclusively focuses on the steady-state performance of these systems. This includes their steady-state expected waiting times and matching probabilities among customer classes and servers. Deriving heavy traffic limits to study the transient behaviour of these service systems and their stationary distributions, as explored in many of the papers we review subsequently, represents a more ambitious goal that lies beyond the scope of this paper.

A range of questions can be answered using heavy traffic approximations. In the context of parallel service systems, Harrison and Lopez [15] study the question of optimal control of parallel service systems, that is, which servers should be used to serve which customer classes, and in which order should the different customer classes be served. Harrison and Lopez [15] solve an approximating Brownian control problem, and conjecture that a discrete review policy will minimize 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 [14]. Williams [31] and Bell and Williams [5] go on to prove the asymptotic optimality of a continuous review policy for a two-server system. Following this work, Mandelbaum and Stolyar [23] 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* (CRP) condition on the connectivity and/or compatibility between customer classes and servers (see [15]). Roughly speaking,[3] 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. [9]. There has already been some work moving beyond the complete resource pooling assumption. Kushner and Chen [20] 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 [25] generalizes Harrison and Lopez [15] 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 [27] and Hurtado Lange and Maguluri [16]. Shah and de Veciana [27] look at a system in which servers simultaneously work to process the same job, while Hurtado Lange and Maguluri [16] analyse a generalized 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 [28] looks at the problem of calculating matching rates for a parallel service system operating under

Footnote 2 continued
be adopted. 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.

[3] A precise definition of complete resource pooling in the context of our work is given in Definition 3.

FCFS, that is, with what probability is each customer 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 [7], a related approximation based on Ohm's law proposed by Fazel-Zarandi and Kaplan [10] and a quadratic programming formulation proposed by Afèche et al. [3]. Of these papers looking at the performance of parallel service systems under FCFS, Afèche et al. [3] is the only one to also look at calculating waiting times as we do here. Another contribution of Afèche et al. [3] is to study the question of the design of matching topologies fixing the scheduling policy. While Afèche et al. [3] studies this design question for a FCFS service discipline, Varma and Maguluri [29] studies the same question of the design of matching topologies under a MaxWeight service discipline.

The specific model we look at here is a generalization of Afèche et al. [3], 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 [26] and Green [12], who look at the steady-state performance of these systems given a particular hierarchical compatibility structure between customer classes and service classes, and Kaplan [17, 18], who similarly analysed the steady-state performance of parallel queuing systems, but for more general compatibility structures. Following Kaplan [17, 18], Kaplan's multi-class multi-server queueing model was adapted by Caldentey and Kaplan [7], who introduced an infinite-bipartite matching model to analyse matching probabilities under a FCFS service discipline. The model of Caldentey and Kaplan [7] was further developed by Caldentey et al. [8] and then adapted by Adan and Weiss [2] to that of a multi-class multi-server parallel queuing system, which is the model we use here.

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. [6], Mairesse and Moyal [22], and Moyal and Perry [24] 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. [8] were able to use a particular Markov chain representation to calculate the steady-state distribution of the matching system for particular classes of matching topologies. Adan and Weiss [1] 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 [2] 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. [3] 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 [11] for a comprehensive review of related papers and models).

The rest of the paper is organized as follows. In Sect. 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 analyse the performance of the system. Section 3 is devoted to the derivation of the limiting steady-state waiting times of the different customer 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 Sect. 4, we study the steady-state matching probabilities between customer 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 Sect. 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 customer 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 6 contains the proofs and additional discussion of our main results Theorems 1 and 2. Some concluding remarks and possible directions in which our work can be extended are present in Sect. 7. Finally, the Appendix contains additional proofs of various intermediate results.

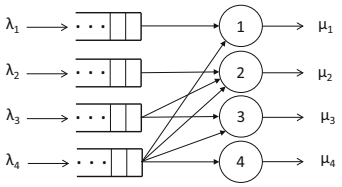
2 Model description

In this section, we provide a detailed mathematical description of the model and basic definitions. To simplify our notation, we will adopt the following conventions throughout the paper. For a positive integer k, $[k] := \{1, 2, \ldots, k\}$. All vectors are column vectors, and for a vector $x \in \mathbb{R}^k$, $\langle x \rangle := \sum_{i \in [k]} x_i$.

We consider a service system as follows. We have a set of m servers organized into a set of n customer classes. Each customer 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 customer class i can be served by server j iff $m_{ij} = 1$. We will also refer to the compatibility matrix M as the *menu of customer classes* or simply the *menu*. Customers arrive to the customer classes according to independent Poisson processes. We let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be the arrival rates into the different customer classes. Service times are exponentially distributed, and depend only on the server. The vector of service rates will be denoted by $\mu = (\mu_1, \ldots, \mu_m)$. Servers will serve customers they are compatible with according to a FCFS-ALIS service discipline.

Example To illustrate, Fig. 1 depicts an example of a queueing system with four servers ($m = 4$) and four customer classes ($n = 4$), and its corresponding matching menu M.

In this example, class 1 is compatible with server 1; class 2 is compatible with server 2; class 3 is compatible with servers 2 and 3; and class 4 is compatible with all servers. Note that a server may belong to multiple customer classes. □



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

Fig. 1 Example of a queueing system with four customer classes and four servers

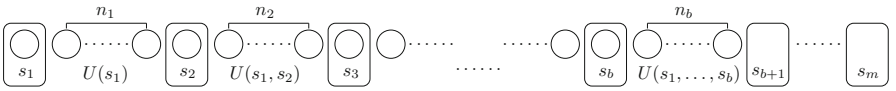


Fig. 2 A general state $x = (s_1, n_1, s_2, n_2, \ldots, s_b, n_b, s_{b+1}, \ldots, s_m)$ of the Markov chain

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

We are only interested in systems which operate with stable queue lengths. The following result, from [2] tells us exactly which triplets (λ, μ, M) produce stable steady-state outcomes.

Proposition 1 [2, Theorem 2.1] *For a menu M with arrival rates λ and service rates μ, define the slack of a set of servers $\Delta_M(\mathcal{S})$ for $\mathcal{S} \subseteq [m]$ as*

$$\Delta_M(\mathcal{S}) := \sum_{j \in \mathcal{S}} \mu_j - \sum_{i \in U(\mathcal{S})} \lambda_i \quad \text{for all } \mathcal{S} \subseteq [m], \tag{2}$$

where $U(\mathcal{S}) := \left\{ i \in [n] \colon \sum_{j \in \mathcal{S}^c} m_{ij} = 0 \right\}$ is the subset of customer classes that can only be served by servers in $\mathcal{S}$. The menu M admits a steady state under a FCFS-ALIS service discipline if and only if $\Delta_{\mathcal{S}}(M) > 0$ for all $\mathcal{S} \subseteq [m]$.

It is often clear from context which menu M the slacks are being defined for, in which case we drop the M from the notation.

2.1 Steady state results for fixed arrival rates

Our results build on the steady-state analysis of a carefully crafted Markov chain representation of the system proposed by [2]. A state in this Markov chain is described by three components: (i) a permutation of servers $s = (s_1, \ldots, s_m)$, (i) 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. A generic state x is given by a tuple $x = (s_1, n_1, s_2, n_2, \ldots, s_b, n_b, s_{b+1}, \ldots, s_m)$, as illustrated in Fig. 2.

Each circle represents a customer in the system, ordered from left to right based on their arrival times, with the leftmost customer being the oldest. The boxes represent the servers. Those containing a customer (circle) are the busy servers (i.e. servers s_1 to s_b), while the rest are idle servers (i.e. servers s_{b+1} to s_m). Idle servers are ordered

from left to right based on the duration of their idleness, with server s_{b+1} being idle for the longest period. The number of customers in the queue who arrived after the customer being served by server ℓ but before the customer being served by server $\ell+1$ is denoted by n_ℓ, for $\ell = 1, \ldots, b$. Due to the FCFS-ALIS service discipline, we know these customers are only compatible with servers in the set $(s_1, \ldots, s_\ell)$. This implies that each of these n_ℓ customers must belong to a customer class within $U(s_1, \ldots, s_\ell)$ and is incompatible with any of the servers in $s_{\ell+1}, \ldots, s_m$.

According to [2, Theorem 2.1], the steady-state probability of state x admits the product form:

$$\pi(x) = \mathcal{B} \prod_{\ell=1}^{b} \frac{\lambda_{U(s_1,\ldots,s_\ell)}^{n_\ell}}{\mu_{\{s_1,\ldots,s_\ell\}}^{n_\ell+1}} \prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1}, \tag{3}$$

where $\mathcal{B}$ is an appropriate normalizing constant, and $C(\mathscr{S}) = \left\{i \in [n] \colon m_{ij} = 1 \text{ for some } j \in \mathscr{S}\right\}$ is the set of all customer classes that can be served by some server in $\mathscr{S} \subseteq [m]$. Additionally, each of the n_ℓ customers 'between' server s_ℓ and server $s_{\ell+1}$ belongs to customer 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 Fig. 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

$$\mathbb{E}(n_4 | x \in X_{(s_1,s_2,s_3,s_4)}) = \frac{\mathcal{B} \cdot \langle\lambda\rangle \langle\mu\rangle}{(\mu_1 - \lambda_1)(\mu_1 + \mu_2 - (\lambda_1 + \lambda_2))(\langle\mu\rangle - \mu_4 - (\langle\lambda\rangle - \lambda_4))(\langle\mu\rangle - \langle\lambda\rangle)}, \tag{4}$$

where $\langle\lambda\rangle := \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4$, $\langle\mu\rangle := \mu_1 + \mu_2 + \mu_3 + \mu_4$ 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 $\frac{\lambda_4}{\langle\lambda\rangle}\mathbb{E}[n_4 | x \in X_{(s_1,s_2,s_3,s_4)}]$.

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 Heavy traffic scaling

We consider a conventional heavy traffic regime in which the arrival rates approach the capacity of the service system from below, while the number of customer classes and servers, and the service menu remain constant. We parameterize our systems by ϵ, and let the service system approach heavy traffic as $\epsilon \downarrow 0$. Specifically, we assume there exist two vectors $\Lambda \in \mathbb{R}^n_+$ and $\gamma \in \mathbb{R}^n$ independent of ϵ so that the vector of arrival rates in the ϵth system is given by

$$\lambda_i{}^{(\epsilon)} = \Lambda_i - \gamma_i \epsilon + o(\epsilon) \geq 0 \quad \text{for all } i \in [n] \text{ and } 0 < \epsilon < \epsilon_+, \tag{5}$$

for some some $\epsilon_+ > 0$. The vector Λ is the limiting vector of arrival rates while the vector γ captures the direction of convergence to heavy traffic. In what follows, we assume that the queueing system satisfies the following assumption.

Assumption 1 The inputs of the queueing system (Λ, μ, γ) satisfy:

(i) $\langle \lambda \rangle = \langle \mu \rangle$,
(ii) $\langle \gamma \rangle > 0$,
(iii) $\gamma_i < 0$ for all $i \in [n]$ such that $\Lambda_i = 0$.

Parts (i) and (ii) ensure that for the sequence of arrival rates $\lambda^{(\epsilon)}$ in (5) the system approaches heavy traffic from below. Part (iii) is implied by $\lambda_i{}^{(\epsilon)} > 0$ for all $0 < \epsilon < \epsilon_+$, but we include it in Assumption 1 for clarity. Note that for $i \in [n]$ such that $\Lambda_i > 0$, we allow γ_i to be positive, negative, or zero.

It is worth mentioning that Afèche et al. [3] considered a heavy traffic scaling that is a special case of (5) in which $\gamma = \Lambda$, that is, the proportions of customers of different types remain constant as the system approaches heavy traffic. Additionally, Afèche et al. [3] requires that $\Lambda_i > 0$ for all $i \in [n]$. We relax that assumption here and allow for customer classes with $\Lambda_i = 0$ and $\gamma_i < 0$. Such classes, with vanishing arrival rate in the heavy traffic limit, might be relevant for 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*) Consider a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 and let $\lambda^{(\epsilon)}$ be given by (5). A menu M is *admissible* if there exists an $\epsilon_+ > 0$ such that for all $0 < \epsilon < \epsilon_+$ and $\mathscr{S} \subseteq [m]$ the following conditions are satisfied:

$$\text{(i)} \quad \Delta_M^{(\epsilon)}(\mathscr{S}) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U(\mathscr{S})} \lambda_i^{(\epsilon)} > 0 \qquad \text{and} \qquad \text{(ii)} \quad \Delta_M^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon). \tag{6}$$

We let $\mathcal{M}(\Lambda, \mu, \gamma)$ denote the set of all menus M that are admissible for the queueing system with inputs (Λ, μ, γ).

In (6) part (ii), the Big Omega notation $f(\epsilon) = \Omega(\epsilon)$ stands for $\limsup_{\epsilon \to 0} \frac{|f(\epsilon)|}{\epsilon} > 0$.

In words, Definition 1 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 customer classes converge when scaled by ϵ. It is worth noting that the set $\mathcal{M}(\Lambda, \mu, \gamma)$ of admissible menus is non-empty for all triplets (Λ, μ, γ) satisfying Assumption 1. To see this, observe that the complete menu M such that $m_{ij} = 1$ for all $i \in [n]$ and $j \in [m]$ is admissible for all (Λ, μ, γ) satisfying Assumption 1. The complete menu will operate like a single queue with arrival rates $\langle \lambda^{(\epsilon)} \rangle$ that is served by all servers.

3 Mean waiting times in heavy traffic

We are interested in calculating the mean waiting times of the different customer 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 instead look at the scaled mean waiting time

$$\widehat{W}_i^{(\epsilon)} = \epsilon \cdot W_i^{(\epsilon)}, \tag{7}$$

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. [3].

3.1 Feasible flows and complete resource pooling

We begin by identifying the feasible flows of customers between customer classes and servers. For a menu M, vector of arrival rates λ, and service capacities μ, we define the set of feasible flows as:

$$\mathcal{F}(\lambda, \mu, M) := \left\{ f = [f_{ij}] \geq 0 : \textstyle\sum_{i \in [n]} f_{ij} \leq \mu_j, ; \ \sum_{j \in [m]} f_{ij} = \lambda_i, \ ; \ f_{ij} = 0 \text{ if } m_{ij} = 0 \right\}. \tag{8}$$

If $M \in \mathcal{M}(\Lambda, \mu, \gamma)$ is an admissible menu then there exists an $\epsilon_+ \in \mathbb{R}_+$ such that $\mathcal{F}(\lambda^{(\epsilon)}, \mu, M)$ is non-empty for all $0 < \epsilon < \epsilon_+$. The following lemma shows that $\mathcal{F}(\Lambda, \mu, M)$ is also non-empty.

Lemma 1 *Consider a system with inputs (Λ, μ, γ) satisfying Assumption 1 and let $M \in \mathcal{M}(\Lambda, \mu, \gamma)$ be an admissible menu. Then, the set $\mathcal{F}(\Lambda, \mu, M)$ is non-empty. Furthermore, every sequence of flows $f^{(\epsilon)} \in \mathcal{F}(\lambda^{(\epsilon)}, \mu, M)$ has a sub-sequence that converges to some $\tilde{f} \in \mathcal{F}(\Lambda, \mu, M)$.*

Proof The proof of this and other results are relegated to the Appendix unless otherwise stated. □

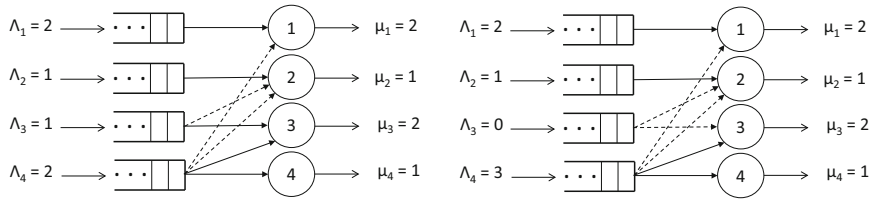
As this lemma suggests, the set $\mathcal{F}(\Lambda, \mu, M)$ contains information about what sort of flows it is possible to observe in heavy traffic. We will use the set of feasible limiting flows to determine which servers have a positive probability of serving which customer classes in the limit.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

Definition 2 (*Residual Matching*) For a system with inputs (Λ, μ, γ) satisfying Assumption 1 and an admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$, 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 $\tilde{f} \in \mathcal{F}(\Lambda, \mu, M)$ such that $\tilde{f}_{ij} > 0$.

Intuitively, for a customer class i and server j with $m_{ij} = 1$ but $\breve{m}_{ij} = 0$, the flow of customers from customer class i to server j must vanish in the heavy traffic limit. Afèche et al. [3] 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 Fig. 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 $\mathcal{F}(\Lambda_a, \mu, M)$ and $\mathcal{F}(\Lambda_b, \mu, M)$, given by

$$f^a_{ij} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad f^b_{ij} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix}. \tag{9}$$

In example (a), the arcs in the compatibility network with $m_{ij} = 1$ and $\breve{m}_{ij} = 0$ are (3,2), (4,1) and (4,2). While customer 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. Similarly, while customer class 3 is compatible with server 2, there will be zero flow between them 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) of Fig. 3, 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 arcs (3,2) and (3, 3) with $m_{32} = m_{33} = 1$ and $\breve{m}_{32} = \breve{m}_{33} = 0$. In



(a) Residual matching (a) (b) Residual matching (b)

Fig. 3 Examples of residual matchings

panel (b) of Fig. 3 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 servers 2 and 3.

Knowing the residual matching allows us to decompose the initial bipartite matching system into a partition of independent components, which Afèche et al. [3] refer to as *complete resource pooling* (CRP) components.

Definition 3 (*CRP Component*) For a system with inputs (Λ, μ, γ) satisfying Assumption 1 and an admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$, 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 customer classes and servers forms a *complete resource pooling (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 customer classes $\mathcal{C}_k$ and the subset of servers $\mathcal{S}_k$ that belong to $\mathbb{C}_k$. In Fig. 3, the queueing system in panel (a) has three CRP components $\mathbb{C}_1 = (\mathcal{C}_1, \mathcal{S}_2) = (\{1\}, \{1\})$, $\mathbb{C}_2 = (\mathcal{C}_2, \mathcal{S}_2) = (\{2\}, \{2\})$, and $\mathbb{C}_3 = (\mathcal{C}_3, \mathcal{S}_3) = (\{3, 4\}, \{3, 4\})$. We will use $k(i)$ and $k(j)$ to denote the component that customer class i or server j is part of, where the use should be clear from context. Also, with a slight abuse of notation, we denote the aggregate arrival and service rates for the CRP components under $\lambda^{(\epsilon)}$ as:

$$\forall k \in [K] \,:\, \widetilde{\lambda}_k^{(\epsilon)} = \sum_{i \in \mathcal{C}_k} \lambda_i{}^{(\epsilon)} =: \widetilde{\Lambda}_k - \epsilon \widetilde{\gamma}_k + o(\epsilon), \quad \text{and} \quad \widetilde{\mu}_k = \sum_{j \in \mathcal{S}_k} \mu_j, \tag{10}$$

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 ϵ. While each CRP component is critically loaded, the "well-connectedness" within a CRP component allows shifting load from one customer class to another on short time scales. In particular, we will show in Theorem 1 that under a FCFS-ALIS policy, waiting times are balanced in such a way that customer classes that belong to the same CRP component have the same limiting scaled mean waiting time in the heavy traffic limit.

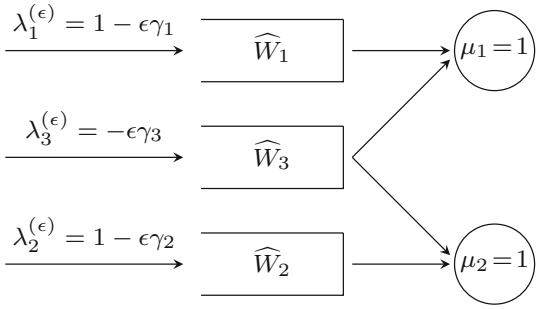


Fig. 4 Example of queueing system with a customer class with zero limiting arrival rate

It is worth noting that we allow for customer classes with no arrivals in the heavy traffic limit, that is $\Lambda_i = 0$ (e.g., customer class 3 in Fig. 3 panel (b)). Each of these customer classes with $\Lambda_i = 0$ forms a separate CRP component with an empty server set. We denote this subset of such CRP components by $\mathcal{I}_0 := \{k : \Lambda_k = 0\}$ and by $K' := K - |\mathcal{I}_0|$ the number of CRP components with non-empty sets of servers. For notational convenience, we index the CRP components so that $\widetilde{\Lambda}_k > 0$ for $k = 1, \ldots, K'$.

Given our interest in the system's performance in the heavy traffic limit, as $\epsilon \downarrow 0$, it might be tempting to disregard the existence of customer classes in $\mathcal{I}_0$ with a zero limiting arrival rate. However, these classes have positive arrival rates in the pre-limit, and this fact significantly affects the behaviour of the heavy traffic limit. Let us consider a simple example to illustrate this point.

Example Consider a queueing system with three customer classes and two servers as depicted in Fig. 4 with $\gamma_1 > 0$, $\gamma_2 > 0$ and $\gamma_3 < 0 < \gamma_1 + \gamma_2 + \gamma_3$, so that the conditions in Assumption 1 are satisfied. Customer class 3 has a zero arrival rate in the heavy traffic limit.

If we were to remove this class then the queueing system would reduce to two independent M/M/1 queues and the scaled waiting times would be equal to $\widehat{W}_i = 1/\gamma_i$ for $i = 1, 2$. However, as we will demonstrate in Theorem 1, the actual limiting scaled waiting times for these classes are $\widehat{W}_i = 1/\gamma_i + 1/\langle\gamma\rangle - 1/(\langle\gamma\rangle - \gamma_3)$ for $i = 1, 2$. These waiting times are higher (given that $\gamma_3 < 0$) than those obtained by neglecting the existence of customer class 3. □

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. Each node in the DAG corresponds to a CRP component. There is an arc in the DAG from a CRP component $\mathcal{C}_{k_1}$ to a component $\mathcal{C}_{k_2}$ if there is a customer class in $\mathcal{C}_{k_1}$ that can be served by a server in $\mathcal{C}_{k_2}$ in the original menu M. This DAG is useful as it defines a precedence relation among customer classes. Since there is a customer class in $\mathcal{C}_{k_1}$ that can be served by a server in $\mathcal{C}_{k_2}$, component $\mathcal{C}_{k_1}$ can "off-load" its customers to the servers of component $\mathcal{C}_{k_2}$. This means the waiting time of customer classes in

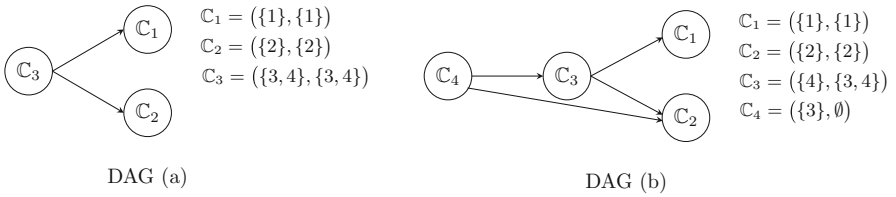


Fig. 5 Examples of DAGs

component $\mathcal{C}_{k_1}$ cannot exceed that of customer classes in component $\mathcal{C}_{k_2}$ 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 $\{\mathbb{C}_k = (\mathcal{C}_k, \mathcal{S}_k) : k \in [K]\}$ induced by the residual matching $\breve{M}$, we define the directed acyclic graph $\mathcal{D} = ([K], \mathcal{A})$ associated with M as follows: the 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 customer class $i \in \mathcal{C}_{k_1}$ and a server $j \in \mathcal{S}_{k_2}$ such that $m_{ij} = 1$.[4]

Returning to our examples in Fig. 3, the DAGs are given as follows.

In both cases, customer 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 customer class 3 but no servers, since customer class 3 has an arrival rate of 0. Therefore $\mathbb{C}_4$ has directed arcs to $\mathbb{C}_2$ and $\mathbb{C}_3$, as these are the CRP components containing the servers that customer class 3 is compatible with.

As we mentioned earlier, our computations for the heavy traffic waiting times build on the work of [2]. 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. [3] for the less general scaling, in heavy traffic this entails restricting attention to only certain permutations of the CRP components which have asymptotically nonzero steady-state probability. We show in Proposition 7 below that this also holds for our more general scaling. The topological orders of the DAG $\mathcal{D}$ provide these permutations. We begin by considering the topological orders of the DAG $\mathcal{D}$ restricted to only those CRP components with $\widetilde{\Lambda}_k > 0$, since those are the CRP components that contain servers. The following definition is analogous to Definition 6 in Afèche et al. [3].

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 $(k_1, k_2) \in \mathcal{A}$, 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 σ of $[K']$ that induce a topological order on components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$.

[4] Afèche et al. [3, Lemma 2] formally proves that the directed graph in this definition is in fact acyclic.

Returning to our examples in Fig. 5, 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 $\mathbb{C}_1$ 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 $(\mathbb{C}_2, \mathbb{C}_1, \mathbb{C}_3)$.

We now consider those CRP components with $\widetilde{\Lambda}_k = 0$. Each of these components consists of exactly one customer class and no servers. This means these CRP components have no incoming arcs in the DAG $\mathcal{D}$, and can only have a directed arc pointing to CRP components with non-empty server sets. In the next definition, we will expand the idea of topological orders to include these server-less CRP components, by creating ordered partitions of CRP components in the following way: for each permutation $\sigma \in \mathcal{T}(\mathcal{D}, K')$, we associate each server-less CRP component with the CRP component that is reachable from it in the DAG that has the shortest steady-state wait, or in other words the reachable CRP component that comes last in the topological order.

Definition 6 (*Ordered Partitions of CRP Components*) For every DAG $\mathcal{D}$ on a collection of CRP components, and for each $\sigma \in \mathcal{T}(\mathcal{D}, K')$, we partition the indices of the CRP components $[K]$ by associating a subset for each $k \in [K']$ as follows:

$$\mathsf{comps}(\sigma, k) := \{\sigma(k)\} \cup \{\kappa \in \mathcal{I}_0 : k = \max\{k' \in [K'] : \ (\kappa, \sigma(k')) \in \mathcal{A}\}\}. \quad (11)$$

In words, for each index $k \in [K']$, we create a set containing the index $\sigma(k)$, as well as all indices of all CRP components $\mathbb{C}_\kappa$ with $\widetilde{\Lambda}_\kappa = 0$ for which the component $\mathbb{C}_{\sigma(k)}$ is the last component in the topological order induced by σ that $\mathbb{C}_\kappa$ is connected to with a directed arc.

For any permutation $\sigma \in \mathcal{T}(\mathcal{D}, K')$ and CRP component index $k \in K$, we will use the shorthand $\mathsf{comps}^{-1}(\sigma, k)$ to denote the index $k' \in [K']$ such that $k \in \mathsf{comps}(\sigma, k')$.

Returning again to the example in Fig. 5, example (a) has no CRP components such that $\widetilde{\Lambda}_k = 0$, and so for each σ and each k, $\mathsf{comps}(\sigma, k)$ is the set containing the index of the CRP component at position k of the permutation σ. In example (b), $\mathbb{C}_4$ has $\widetilde{\Lambda}_4 = 0$, so for each permutation $\sigma \in \mathcal{T}(\mathcal{D}, K')$, we need to determine for which index $k \in K'$ we have $4 \in \mathsf{comps}(\sigma, k)$, or in other words, what the value of $\mathsf{comps}^{-1}(\sigma, 4)$ is. CRP component $\mathbb{C}_4$ has directed arcs to both $\mathbb{C}_2$ and $\mathbb{C}_3$. So for each topological order induced by permutations $\sigma \in \mathcal{T}(\mathcal{D}, K')$, we look at which component out of $\mathbb{C}_2$ and $\mathbb{C}_3$ has the later position. Recall the two permutations in $\mathcal{T}(\mathcal{D}, K')$ are $\sigma_1 = (1, 2, 3)$ and $\sigma_2 = (2, 1, 3)$. In both permutations, $\mathbb{C}_3$ comes after $\mathbb{C}_2$. Thus in the ordered partitions of CRP components generated by σ_1 and σ_2, $\mathbb{C}_4$ is in the same set as $\mathbb{C}_3$. So for both permutations σ_1 and σ_2, $\mathsf{comps}(\sigma_1, 3) = \mathsf{comps}(\sigma_2, 3) = \{3, 4\}$.

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 asso-

ciated function $\mathsf{comps}(\sigma_t, \cdot)$ defined in (11), we define the unnormalized probability of being in a state associated with the topological order σ_t as:

$$\mathbb{Q}(\sigma_t) = \prod_{\kappa \in [K']} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t, \ell)}}, \tag{12}$$

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 σ_t as:

$$w_{\sigma_t,k} = \sum_{\kappa = \mathsf{comps}^{-1}(\sigma_t,k)}^{K'} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)}}. \tag{13}$$

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

Lemma 2 *For a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 and some admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$, we have that for all permutations $\sigma_t \in \mathcal{T}(\mathcal{D}, K')$ of CRP components $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$ and for all $\kappa \in [K']$, $\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t,\ell)} > 0$.*

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 customer classes.

Theorem 1 *For a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 and some admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$, let $\breve{M}$ be the residual matching and $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ be the collection of CRP components induced by $\breve{M}$. Then, customer 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*

$$\widehat{W}_{\mathbb{C}_k} = \sum_{t=1}^{T(M)} \left(\frac{\mathbb{Q}(\sigma_t)}{\mathbb{Q}(\sigma_1) + \mathbb{Q}(\sigma_2) + \cdots + \mathbb{Q}(\sigma_{T(M)})} \right) w_{\sigma_t,k}, \tag{14}$$

where $T(M)$ is the total number of topological orders for the menu M.

The proof of Theorem 1 can be found in Sect. 6.1.

4 Matching probabilities in heavy traffic

Another performance metric of interest is the matching probabilities, that is, for each customer class i and server j, the probability that a customer who joins class i is served by server j. Take any system inputs (Λ, μ, γ), satisfying Assumption 1, and any admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$. Also take any sequence of arrival rates $\lambda^{(\epsilon)}$

satisfying Eq. 5. 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. The results in this section will be limited to systems in which all customer classes have strictly positive arrival rates.

Before stating our results regarding matching probabilities, it will be useful to gain a better understanding of how the admissibility of menus relates to the limiting arrival rates Λ. The following proposition will help develop this understanding

Proposition 2 *Consider any queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 such that $\Lambda_i > 0$ for all $i \in [n]$. Then $\mathcal{M}(\Lambda, \mu, \gamma) \subseteq \mathcal{M}(\Lambda, \mu, \Lambda)$. Further, for any menu M, if $M \in \mathcal{M}(\Lambda, \mu, \Lambda)$, then the menu $\breve{M}$ given by the residual matching of M with limiting arrival rates Λ is also in $\mathcal{M}(\Lambda, \mu, \Lambda)$.*

This tells us that for (Λ, μ, γ) satisfying Assumption 1, a menu M being admissible for model primitives (Λ, μ, Λ), is a necessary condition for M to be admissible for (Λ, μ, γ).

We are now ready to present our first result regarding matching probabilities, which 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 γ, the matching probabilities depend only on the limiting arrival rates. The proof of Theorem 2 and Theorem 1 can be found in Sect. 6.2.

Theorem 2 *Take any limiting arrival rates Λ and service rates μ such that $\langle \Lambda \rangle = \langle \mu \rangle$, and $\Lambda_i > 0$ for all $i \in [n]$. Consider any menu $M \in \mathcal{M}(\Lambda, \mu, \Lambda)$. Also take any two vectors of slacks γ_a and γ_b such that (Λ, μ, γ_a) and (Λ, μ, γ_b) satisfy Assumption 1, and $M \in \mathcal{M}(\Lambda, \mu, \gamma_a)$ and $M \in \mathcal{M}(\Lambda, \mu, \gamma_b)$. Then for any two sequences of arrival rates $\lambda_a^{(\epsilon)}$ and $\lambda_b{}^{(\epsilon)}$ satisfying Eq. 5 with γ_a and γ_b respectively,*

$$\lim_{\epsilon \to 0} p_{ij}^{(\epsilon)}(M, \lambda_a^{(\epsilon)}, \mu) = \lim_{\epsilon \to 0} p_{ij}^{(\epsilon)}(M, \lambda_b^{(\epsilon)}, \mu) \text{ for all } i \in [n] \text{ and } j \in [m].$$

Theorem 2 lets us talk about the matching probabilities of a menu M just in terms of the limiting arrival rates Λ and service rates μ.

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

Corollary 1 *Take any limiting arrival rates Λ and service rates μ such that $\langle \Lambda \rangle = \langle \mu \rangle$ and $\Lambda_i > 0$ for all $i \in [n]$, and take any $M \in \mathcal{M}(\Lambda, \mu, \Lambda)$. Let $\breve{M}$ be the residual matching, and let $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ be the collection of CRP components induced by $\breve{M}$. Then for any customer class $i \in \mathcal{C}_k$ and server $j \in \mathcal{S}_k$,*

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

Theorem 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 Sect. 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. Theorem 1 tells us that despite this, the limiting matching probabilities of menus M and M' are the same.

5 Implications on system design

Before getting into the proofs of our main results, namely Theorems 1 and 2, let us first discuss in this section some of the implications of our heavy traffic analysis in the context of designing and analysing multi-class multi-server queueing systems. Along the way, we will also use this discussion to highlight some key differences between the behaviours of our model and the model presented in Afèche et al. [3].

For the sake of clarity in our exposition, we will limit our discussion in this section to queueing systems (Λ, μ, γ) that satisfy $\Lambda > 0$ as well as the conditions outlined in Assumption 1. That is, systems that do not include any customer classes with zero limiting arrival rate. Consequently, $\mathcal{I}_0 = \emptyset$, $K' = K$, and $\mathsf{comps}(\sigma, \ell) = \sigma(\ell)$ for any topological order σ.

5.1 Menu design

An important objective for service providers managing queueing systems, such as those studied in this paper, is to identify service menus that will achieve good performance in terms of waiting time delays. Depending on the context, the objective may be to minimize the average delay across all customer classes or to minimize the maximum expected delay for any customer class. The following result provides a lower bound on the achievable waiting times that can be implemented.

Proposition 3 *Consider a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1. Consider an admissible matching $M \in \mathcal{M}(\Lambda, \mu, \gamma)$ and let $\mathcal{D} = ([K], \mathcal{A})$ be its associated DAG with CRP components $\{\mathbb{C}_1, \ldots, \ldots, \mathbb{C}_K\}$. Then, $\widehat{W}_{\mathbb{C}_k} \geq \frac{1}{\langle\gamma\rangle}$ for all $k \in [K]$. Furthermore, $\widehat{W}_{\mathbb{C}_k} = \frac{1}{\langle\gamma\rangle}$ for some $k \in [K]$ if and only if on $\mathcal{D}$ there exists a directed path from $\widehat{W}_{\mathbb{C}_k}$ to any other CRP component $\mathbb{C}_\kappa$ with $\kappa \in [K]\backslash\{k\}$. This condition is trivially satisfied if there is only one CRP component.*

A significant implication of the previous result is that under heavy traffic conditions, any service menu inducing a single CRP component-thereby achieving complete resource pooling-will ensure that all customer classes experience the minimum possible expected delay. Consequently, if the service provider aims to minimize waiting time delays and has full flexibility in choosing the service menu, selecting a menu that induces a single CRP component would be optimal. While many menus could satisfy the objective of inducing a single CRP component-for instance, a fully connected menu where each customer class is connected to every server—some menus might be more preferable from a practical standpoint. This is because they could allow for a

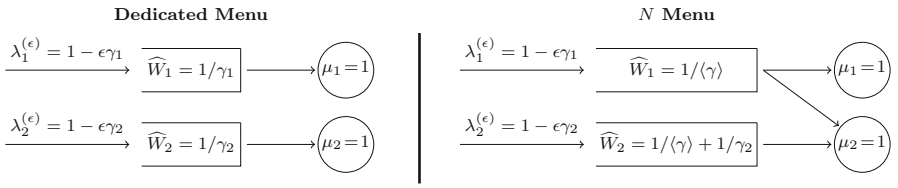


Fig. 6 The dedicated and N menus. $\widehat{W}_i$ is the scaled mean waiting time for customer class $i = 1, 2$

more selective matching of customers to servers. The following proposition is useful in identifying such a menu.

Proposition 4 *Consider a queueing system with inputs* (Λ, μ, γ) *satisfying Assumption 1. Then, any menu* M *such that*

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

is admissible and induces a single CRP component.

Interestingly, Proposition 4 tells us that we do not need to know the values for the slacks γ to design a delay minimizing menu, making it easier to implement in practice.

Menus with a predetermined set of CRP components While a menu inducing a single CRP component minimizes delays, offering such a menu might not be desirable or feasible due to real-world compatibility constraints that dictate which servers can serve which customer types. Motivated by these practical considerations, let us next explore how to select an admissible menu M whose associated DAG $\mathcal{D} = ([K], \mathcal{A})$ induces a given set of CRP components $\mathbb{C}_1, \ldots, \mathbb{C}_K$. Alternatively, we can reframe this question as the problem of identifying a DAG within a given collection of CRP components that leads to minimal average waiting time delays.

In the process of answering this question one can be inclined to believe that adding arcs to a given DAG will reduce expected delays as this will give additional flexibility to a service system. However, in general adding arcs to a DAG may potentially increase, decrease, or not affect the average delays. This is illustrated in the following two-server example.

Example Let (Λ, μ, γ) be a queueing system with two customer classes and two serves such that $\Lambda = \mu = (1, 1)$ and $\gamma = (\gamma_1, \gamma_2) > 0$. Consider the following two alternative service menus: (i) Dedicated Menu in which each server serves exclusively one customer class and (ii) N Menu in which server 1 serves exclusively customer class 1 while server 2 serves both customer classes.

Given the scaled mean waiting times $\widehat{W}$ for each customer class, as indicated in Fig. 6, the average delay across both customer classes for the Dedicated and N menus are equal to

$$\overline{W}^{\mathrm{D}} = \frac{1}{2}\left(\frac{1}{\gamma_1} + \frac{1}{\gamma_2}\right) \quad \text{and} \quad \overline{W}^{\mathrm{N}} = \frac{1}{\langle\gamma\rangle} + \frac{1}{2\,\gamma_2}, \tag{15}$$

and so the difference in average delays is $\overline{W}^{\mathrm{D}} - \overline{W}^{\mathrm{N}} = \frac{1}{2\gamma_1} - \frac{1}{\langle\gamma\rangle}$. It follows that whether the N menu leads to lower, equal, or higher average delays than the Dedicated menu depends on whether $\gamma_1 < \gamma_2$, $\gamma_1 = \gamma_2$, or $\gamma_1 > \gamma_2$, respectively. In particular, the case $\gamma_1 > \gamma_2$ demonstrates that increasing the connectivity between customer classes and servers in the design of a menu does not necessarily lead to reduced average delays; in some cases, it may actually increase them, analogous to a form of Braess's paradox. Therefore, if a service provider is contemplating adding more flexibility to a system, it is crucial to carefully assess how this flexibility is integrated. □

Based on Theorem 1, we can establish the connection between the average waiting time of a given menu and its underlying DAG of CRP components. Recall that Eq. 14 defined the delay of a given CRP component conditional on a particular topological order. From this, we can similarly define $\overline{w}_\sigma$, the average delay across all customer classes conditional on the topological order σ, as

$$\overline{w}_\sigma = \frac{1}{\langle\mu\rangle} \sum_{\kappa=1}^{K'} \frac{\sum_{k=1}^{\kappa} \widetilde{\mu}_{\sigma(k)}}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\sigma(\ell)}}. \tag{16}$$

As a result, the average expected delay across all customer classes for a particular menu M equals

$$\overline{W} = \sum_{t=1}^{T(M)} \left(\frac{\mathbb{Q}(\sigma_t)}{\mathbb{Q}(\sigma_1) + \mathbb{Q}(\sigma_2) + \cdots + \mathbb{Q}(\sigma_{T(M)})} \right) \overline{w}_{\sigma_t}. \tag{17}$$

Here we can see an important consequence of our model with arbitrary vector of slacks γ and the results in Afèche et al. [3], who consider the special case $\Lambda = \gamma$. Indeed, when $\Lambda = \gamma$ the average expected delay in (17) reduces to $\overline{W} = K/\langle\mu\rangle$ (see [3, Corollary 2]), which depends exclusively on the total service capacity $\langle\mu\rangle$ and total 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.

In Eq. 17 both $\mathbb{Q}(\sigma_t)$ and $\overline{w}_{\sigma_t}$ depends exclusively on the topological order σ_t and not on the DAG $\mathcal{D}$ itself. The dependence of $\overline{W}$ on the $\mathcal{D}$ is reflected in the collection of topological orders $\mathcal{T}(\mathcal{D}, K)$ induced by $\mathcal{D}$. 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 7 Consider a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 with $\Lambda > 0$ and let $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_K\}$ be a given collection of CRP components in this system. We say that a topological order σ is *admissible* for $\mathbb{C}$ if $\sum_{\ell=1}^{k} \widetilde{\gamma}_{\sigma(\ell)} > 0$ for all $k \in [K]$. We let $\Sigma(\mathbb{C})$ denote the collection of all admissible topological orders for $\mathbb{C}$.

The following lemma tells us how admissible topological orders relate to admissible menus and which DAGs are feasible given a particular collection $\mathbb{C}$ of CRP component.

Lemma 3 *Consider a queueing system with inputs (Λ, μ, γ) satisfying Assumption 1 with $\Lambda > 0$ and let $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_K\}$ be a given collection of CRP components in this system. For any any admissible topological order $\sigma \in \Sigma(\mathbb{C})$, we can construct an admissible menu $M \in \mathcal{M}(\Lambda, \mu, \gamma)$ such that the DAG induced by M only admits the topological order σ. Furthermore, if σ is not admissible, then there are no admissible menus M that admit the topological order σ.*

Equipped with Lemma 3, we can formulate the problem of minimizing average delays for a given collection $\mathbb{C}$ of CRP components by identifying the admissible topological order with the lowest condition delays.

Proposition 5 *Consider a queueing system with fixed inputs (Λ, μ, γ) satisfying the conditions in Assumption 1 with $\Lambda > 0$. Consider the class of all menus which induce a given collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_K\}$. Let $\sigma^* := \text{argmin}\{\overline{w}_\sigma : \sigma \in \Sigma(\mathbb{C})\}$ be the admissible topological order with minimum conditional average delay. Then, the DAG that minimizes the average delay across all customer classes is the one that only allows for $\sigma^*(\mathbb{C})$ as its unique topological order.*

Note that in the optimal DAG identified in Proposition 5 there is a direct arc from CRP component $\mathbb{C}_k$ to $\mathbb{C}_\ell$ if and only if $\sigma^*(k) = \sigma^*(\ell) + 1$. That is, the optimal DAG that minimizes the average delay across customer classes is a single path: $\mathbb{C}_{\sigma^*(K)} \to \mathbb{C}_{\sigma^*(K-1)} \to \cdots \to \mathbb{C}_{\sigma^*(2)} \to \mathbb{C}_{\sigma^*(1)}$. Thus, using Eq. (17), we can find σ^* by solving

$$\sigma^* = \text{argmin}_{\sigma \in \Sigma(\mathbb{C})} \frac{1}{\langle \mu \rangle} \sum_{\kappa=k}^{K} \frac{\mu_{\sigma(k)}}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\sigma(\ell)}}.$$

5.2 Perturbation analysis and implementable waiting time delays

In the previous section, we discussed designing a service menu M to minimize customers' average waiting times, considering the system's inputs (Λ, μ, γ) as given. This section extends that discussion to include the possibility of selecting not only the menu M but also the vector γ of slacks. From a practical standpoint, this additional degree of flexibility can be interpreted in the context of a system provider that is able to marginally perturbed the arrival rate of the various customer classes effectively changing the direction of convergence of $\lambda^{(\epsilon)}$ to Λ. As we will see, controlling the values of γ allows for a wider range of outcomes than the proportional scaling used in Afèche et al. [3]. The following definition formalizes what we mean by this.

Definition 8 (*Implementable Waiting Times*) Take limiting arrival rates Λ, service rates μ, 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 $M \in \mathcal{M}(\Lambda, \mu, \gamma)$, and the resulting limiting waiting times $\widehat{W}_{\mathbb{C}_k}$ given by (14) are equal to W_k for all $k \in [K]$.

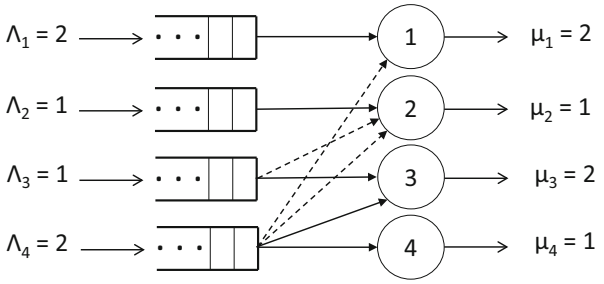


Fig. 7 Queueing system with four customer classes, four servers and three CRP components

Example Let us illustrate the notion of implementability in Definition 8 by revisiting the example in Fig. 3 panel (a) (Fig. 7).

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 $1 \le i \le 4$. We have three CRP components, $\mathbb{C}_1 = (\{1\}; \{1\})$, $\mathbb{C}_2 = (\{2\}; \{2\})$, and $\mathbb{C}_3 = (\{3, 4\}; \{3, 4\})$. From Theorem 1, we obtain the following scaled waiting time delays for each CRP component:

$$\widehat{W}_1 = \frac{1}{\gamma_1} + \frac{1}{\langle\gamma\rangle}, \quad \widehat{W}_2 = \frac{1}{\gamma_2} + \frac{1}{\gamma_1 + \gamma_2 + \gamma_3}, \quad \text{and} \quad \widehat{W}_3 = \frac{1}{\langle\gamma\rangle}.$$

By inspection, we can see that one can implement any vector of delays such that $\min\{\widehat{W}_1, \widehat{W}_2\} > \widehat{W}_3 > 0$. To do this we would let $\gamma_1 = \frac{1}{W_1 - W_3}$, $\gamma_2 = \frac{1}{W_2 - W_3}$, and $\gamma_3 + \gamma_4 = \gamma_1 + \gamma_2 - 1/W_1$.

If we only look at the scaling in Afèche et al. [3], in which $\gamma = \Lambda$, then a single specific vector of waiting times can be implemented. Thus, by allowing γ to change, we increase the set of implementable outcomes. This 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 customer classes. □

As we alluded to in Sect. 3, the DAG provides information about which vectors of waiting times are implementable. The following statement, which is a corollary of Theorem 1, formalizes 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 $W_k \le W_\kappa$ only if $\sigma(\kappa) \le \sigma(k)$.*

Corollary 2 provides a necessary condition for waiting times to be implementable. While completely characterizing the set of implementable waiting times for a particular Λ, μ, 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 9 (*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 $\ell \in [L-1]$.

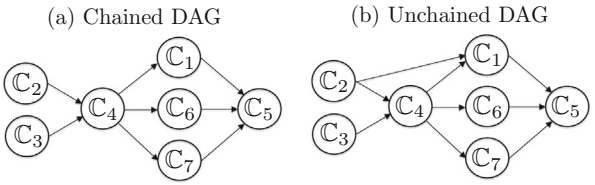


Fig. 8 Examples of chained (**a**) and unchained (**b**) DAGs over seven CRP components

Figure 8 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 $\mathscr{C}_1 = \{\mathbb{C}_2, \mathbb{C}_3\}$, $\mathscr{C}_2 = \{\mathbb{C}_4\}$, $\mathscr{C}_3 = \{\mathbb{C}_1, \mathbb{C}_6, \mathbb{C}_7\}$ 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 9 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 ℓ, 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 6 *Take limiting arrival rates Λ, service rates μ, and a menu M that induces a collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$ and a chained DAG $\mathcal{D} = ([K], \mathcal{A})$ as described in Definition 9. Let $\mathscr{C} = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_L\}$ be the corresponding partition of $\mathbb{C}$. The vector of waiting times $W = (W_1, \ldots, W_K) \in \mathbb{R}_+^K$ is implementable if there exists a vector $\widehat{\mathbb{W}} = (\widehat{\mathbb{W}}_1, \ldots, \widehat{\mathbb{W}}_L) \in \mathbb{R}_+^L$ such that*

(i) $W_k = \widehat{\mathbb{W}}_\ell$ *for all* $k \in [K]$ *such that* $\mathbb{C}_k \in \mathscr{C}_\ell$ *for some* $\ell \in [L]$,
(ii) $\widehat{\mathbb{W}}_\ell < \widehat{\mathbb{W}}_{\ell+1}$ *for* $\ell = 1, \ldots, L-1$.

The following corollary establishes that it is always possible to implement any vector of distinct waiting times by using a simple linear DAG, namely a DAG that induces a single topological order.

Corollary 3 (Linear DAG) *Consider a queueing system with limiting arrival rates Λ, service rates μ and a menu M that induces a collection of CRP components $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$. Let $W = (W_1, \ldots, W_K) \in \mathbb{R}_+^K$ be a given vector of waiting times such that $W_1 > W_2 > \cdots > W_K$. Then, there exists a vector of aggregate slacks $\widetilde{\gamma} = (\widetilde{\gamma}_1, \widetilde{\gamma}_2 \ldots, \widetilde{\gamma}_K) \in \mathbb{R}^K$ such that a linear DAG $\mathcal{D} = ([K], \mathcal{A})$ on $\mathbb{C}$ implements the vector W, where the set of directed arcs is given by $\mathcal{A} = \{(k+1, k) : k = 2, \ldots, K\}$.*

Remark 1 Under the heavy traffic scaling considered in Afèche et al. [3], with $\gamma = \Lambda$, the linear DAG in the previous corollary implements a unique vector of waiting times $W = (W_1, \ldots, W_K)$ with $W_k = \sum_{\kappa=k}^K \frac{1}{\sum_{\ell=1}^\kappa \widetilde{\Lambda}_\ell}$, where $\widetilde{\Lambda}_\ell$ is equal to aggregate slack of CRP component ℓ. □

6 Proof of main results

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, we will recall some notation. In Eq. 10, we defined the aggregate arrival rate for a CRP component $\mathbb{C}_k$ to be $\widetilde{\lambda}_k^{(\epsilon)} = \sum_{i \in \mathcal{C}_k} {\lambda_i}^{(\epsilon)} = \widetilde{\Lambda}_k - \epsilon \widetilde{\gamma}_k + o(\epsilon)$. In Definition 2 for we defined the slack for a subset of servers $\mathscr{S} \subseteq [m]$ as $\Delta(\mathscr{S}) = \mu_{\mathscr{S}} - \lambda_{U(\mathscr{S})}$. In Proposition 1, $U(\mathscr{S})$ is defined as the subset of customer classes that can only be served (or, uniquely served) by servers in $\mathscr{S}$ under the menu M.

We further aggregate the state space described in Sect. 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 $P(s; b)$:

$$\pi(P(s;b)) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_b=0}^{\infty} \mathcal{B} \prod_{\ell=1}^{b} \frac{\lambda_{U(s_1,\ldots,s_\ell)}^{n_\ell}}{\mu_{\{s_1,\ldots,s_\ell\}}^{n_\ell+1}} \prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1} = \frac{\prod_{\ell=b+1}^{m} \lambda_{C(s_\ell,\ldots,s_m)}^{-1}}{\prod_{\ell=1}^{b} \Delta(s_1, \ldots, s_\ell)}. \tag{18}$$

We can use these aggregated states to express the total expected waiting times for each customer class in terms of the aggregated probabilities $\pi(P(s; b))$. The following lemma, which is rephrased from Afèche et al. [3] gives an expression for the mean waiting time for each customer class in terms of the probabilities $\pi(P(s; b))$.

Lemma 4 [3, Lemma 6] *The steady-state mean waiting time of customer 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 Σ_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* (18).

We are able to simplify these expressions further by showing that only a relatively small subset of aggregate states (s, b) have asymptotically nonzero 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 $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}\}$. The following definition formalizes what we mean by this.

Definition 10 (*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 σ.

Returning to our four server example in Fig. 5a, 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 $\sigma_1 = (1, 2, 3)$ and $\sigma_2 = (2, 1, 3)$. Definition 10 tells us the topological order σ_1 induces two possible server permutations, $s_{11} = (s_1 || s_2 || s_3 || s_4)$ and $s_{12} = (s_1 || s_2 || s_4 || s_3)$.

In Proposition 7 in "Appendix D", we prove that only states in which all servers are busy and have server permutations that are induced by the topological orders $\mathcal{T}(\mathcal{D}, K') = (\sigma_1, \ldots, \sigma_T)$ have asymptotically nonzero probabilities in heavy traffic. Further, we show that the asymptotic probabilities of these states can be expressed as

$$\lim_{\epsilon \to 0} \pi(P(s;b)) = \mathcal{B}' \cdot \mathbb{Q}(\sigma) \prod_{k=1}^{K'} \theta_k(\mathbf{s}_k), \tag{19}$$

where $\mathcal{B}'$ is a normalization constant, $\mathbb{Q}(\sigma)$ was defined in (12) 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:

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

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 σ, 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 σ and not the full server permutation s.

Lemma 5 *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 customer class $i \in \mathbb{C}_k$,*

$$\lim_{\epsilon\to 0} \epsilon W_i(s;m) = w_{\sigma,k} := \sum_{\kappa=\mathsf{comps}^{-1}(\sigma,k)}^{K'} \frac{1}{\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}. \tag{20}$$

Combining Proposition 7 with Lemmas 4 and 5, the limiting scaled mean waiting time for customer 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[5] we can reduce the above sum to a sum over server permutations induced by topological orders, and where all servers are busy.

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

as in the theorem statement.

6.2 Proof of Theorem 2

Throughout this section, we will take the menu M, limiting arrival rates Λ and service rates μ, and slacks γ to be given, and largely suppress any dependence on M in the

[5] 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.

notation. We will let $\breve{M}$ be the residual matching of the menu M with arrival rates Λ and service rates μ.

Instead of directly working with the matching rates $p_{ij}^{(\epsilon)}$, 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 μ, limiting arrival rates Λ, 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 Sect. 6.1, we introduced the aggregate states $P(s, b)$ for ever $s \in \Sigma_m$ and $b \in [m]$. We will 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 simultaneously. Specifically, for some $k \in [K']$ and some subpermutation $\mathbf{s}_k \in \Sigma_{\mathbf{S}_k}$, we define

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

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 customer 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 probability of server j serving class i 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 pairs i and j that are not in the same CRP component, as we know the limiting service probabilities of customer 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 steady state 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

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

$$= \lambda_i \sum_{r=j(x)}^{m} \left(\prod_{u=j(x)}^{r-1} \frac{\lambda^{n_u}_{\{U(s_1,\ldots,s_u)\cap\overline{C(j)}\}}}{\lambda^{n_u}_{U(s_1,\ldots,s_u)}} \right) \left(\frac{\lambda^{n_r}_{U(s_1,\ldots,s_r)} - \lambda^{n_r}_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}}{\lambda^{n_r}_{U(s_1,\ldots,s_r)} \left(\lambda_{U(s_1,\ldots,s_r)} - \lambda_{\{U(s_1,\ldots,s_r)\cap\overline{C(j)}\}}\right)} \right). \tag{21}$$

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 $q^0_{ij}(x)$, the part of the expression representing a transition outside of the CRP component. We suppress the dependence on ϵ to reduce clutter in the notation. So

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

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

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

$$q^{(\epsilon)}_{ij}(P(s,m)) = \frac{1}{\pi(P(s,m))} \left[\sum_{x\in P(s,m)} \pi(x) q^+_{ij}(x) + \sum_{x\in P(s,m)} \pi(x) q^0_{ij}(x) \right]. \tag{22}$$

The second term in Eq. 22 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 that the second term in this expression will converge to zero, as we prove in Lemma 11 in "Appendix 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 σ. 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 11, we can write $q^{(\epsilon)}_{ij}(P(s,m))$ as

$$q^{(\epsilon)}_{ij}(P(s,m)) = \frac{\lambda_i}{\pi(P(s,m))} \sum_{n_1=0}^{\infty} \cdots \sum_{n_m=0}^{\infty} \mathcal{B} \prod_{\ell=1}^{m} \frac{\lambda^{n_\ell}_{U(s_1,\ldots,s_\ell)}}{\mu^{n_\ell+1}_{\{s_1,\ldots,s_\ell\}}} q^+_{ij}(s_1,n_1,\ldots,s_m,n_m) + o(1). \tag{23}$$

The following notation will be useful in simplifying this expression. Recall from Definition 2 that $\Delta(S) = \mu_S - \lambda_{U_M(S)}$. It will also be useful to define $\Delta_j(S)$ as

$$\Delta_j(S) = \mu_S - \lambda_{\{U_M(S)\cap\overline{C(j)}\}}. \tag{24}$$

This lets us write Eq. 23 as

$$q_{ij}^{(\epsilon)}(P(s,m)) = \frac{\mathcal{B}\lambda_i}{\pi(P(s,m))}\left(\prod_{\ell=m_{k(j)}+1}^{m}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\left(\prod_{\ell=1}^{m_{k(j)}-1}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)$$
$$\times\left(\prod_{\ell=m_{k(j)-1}+1}^{j-1}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)$$
$$\left[\sum_{r=j(s)}^{m_k(j)}\left(\prod_{u=j(s)}^{r}\frac{1}{\Delta_j(s_1,\ldots,s_u)}\right)\left(\prod_{\ell=r+1}^{m_{k(j)}}\frac{1}{\Delta(s_1,\ldots,s_\ell)}\right)\right.$$
$$\left.\times\left(\frac{1}{\Delta(s_1,\ldots,s_r)}-\frac{1}{\Delta_j(s_1,\ldots,s_r)}\right)\right]+o(1), \tag{25}$$

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

This shows us that the limiting values of $\Delta(s_1\ldots,s_\ell)$ are key in understanding $q_{ij}^{(\epsilon)}(P(s,m))$. Lemma 10 tells us that if $\ell = m_\kappa$ for some $\kappa \in [K']$, then

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

For all other values of ℓ, 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 $S = \{s_{m_{\kappa-1}+1},\ldots,s_\ell\}$. Lemma 8 part (ii) implies that

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

We can use these observations to prove the following lemma.

Lemma 6 *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*

$$q_{ij}(P(s,m)) = \lim_{\epsilon\to 0}\left[\frac{\mathcal{B}\lambda_i}{\pi(P(s,m))\epsilon^{K'}}\mathbb{Q}(\sigma)\left(\prod_{\kappa\neq k(j)}\theta_\kappa(s_\kappa)\right)H_{ij}(\mathbf{s}_{k(j)})\right]$$
$$-\lim_{\epsilon\to 0}\left[\frac{\mathcal{B}\lambda_i}{\pi(P(s,m))\epsilon^{K'-1}}\left(\prod_{\kappa\neq k}\frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}\right)\right.$$
$$\left.\left(\prod_{\kappa\neq k(j)}\theta_\kappa(s_\kappa)\right)G_{ij}(\mathbf{s}_{k(j)})+o(1)\right], \tag{26}$$

where θ_κ *and* H_{ij} *only depend on* $\breve{M}$, Λ, *and* μ.

We provide exact definitions of $\{\theta_k : \Sigma_{\mathcal{S}_k} \to \Re^+\}_{k\in[K']}$, $H_{ij} : \Sigma_{\mathcal{S}_k} \to \Re^+$, and $G_{ij} : \Sigma_{\mathcal{S}_k} \to \Re^+$ in the proof of Lemma 6 in "Appendix E". The important thing to notice is that the first line in Eq. 26 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 Eq. 26 will be nonzero. Thus

$$q_{ij}(P(s,m)) = \frac{\mathcal{B}'\lambda_i}{\pi(P(s,m))}\mathbb{Q}(\sigma)\left(\prod_{\kappa\neq k}\theta_\kappa(s_\kappa)\right)H_{ij}(\mathbf{s}_{k(j)}). \tag{27}$$

Using Eq. 27, and the fact that the q_{ij} are matching probabilities and must sum to one, we can rewrite $q_{ij}(P(s,m))$ as

$$q_{ij}(P(s,m)) = \frac{H_{ij}(\mathbf{s}_{k(j)})}{\sum_{i'\in\mathcal{C}_k} H_{i'j}(\mathbf{s}_{k(j)})}. \tag{28}$$

Since Eq. 28 holds for any server permutation $s \in \Sigma$, and the right hand side depends only on $\mathbf{s}_k$ and not on the rest of the server permutation, this implies that

$$q_{ij}(P_{k(j)}(\mathbf{s}_{k(j)})) = \frac{H_{ij}(\mathbf{s}_{k(j)})}{\sum_{i'\in\mathcal{C}_k} H_{i'j}(\mathbf{s}_{k(j)})}. \tag{29}$$

As Lemma 6 states, $H_{ij}(\mathbf{s}_{k(j)})$ does not depend on γ. This means that 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 γ. This is captured in the following lemma.

Lemma 7 *For an admissible service menu M with limiting arrival rates Λ service rates μ, and slacks Γ, 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 $\{\theta_\kappa : \Sigma_{\mathcal{S}_\kappa} \to \Re^+\}_{\kappa\in[K']}$ is a function that depends only on $\breve{M}$, Λ, and μ.

Combining Lemma 7 with Eq. 28, we conclude that the limiting service probabilities $\lim_{\epsilon\to 0} q_{ij}^{(\epsilon)}$ do not depend on the exact values of the slacks γ.

7 Concluding remarks

In this paper, we have 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. [3] for a similar class of systems. In Theorem 1 we have provided a general characterization of the mean steady-state waiting time

delay for each customer 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 customer classes. We have also studied the steady-state matching probabilities among customer 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. 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 minimize expected delays and have complete control over the design of the menu, then they should implement a menu that induces a single CRP component.

Our work points towards several promising research directions. Firstly, we suggest exploring the problem of menu design, which involves determining the customer classes to offer when customers can select which queue to join upon arrival. Caldentey et al. [9] have made some preliminary progress in this area. Another area that deserves further investigation is the relationship between delays and the underlying matching topology in our bipartite queueing system. In Sect. 5.1, 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.

In addition, there are alternative modelling choices that could be worth exploring. For example, while we have focussed on conventional heavy traffic scaling, 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.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Appendix A Section 3 proofs

Proof of Lemma 1: Let us define the set $\mathcal{F}_{\max}$ as

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

For all $\epsilon \in [0, \epsilon_o)$, note that $\mathcal{F}(\lambda^{(\epsilon)}, \mu, M) \subseteq \mathcal{F}_{\max}$. Furthermore, since $\mathcal{F}_{\max}$ is a compact set, 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 $\tilde{f} \in \mathcal{F}(\Lambda, \mu, M)$, 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. □

Before proving Lemma 2, we state some properties of CRP components and topological orders that will be useful in proving the remaining results. This lemma has been slightly modified from [3, Lemma 6].

Lemma 8 *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 for limiting arrival rates Λ. 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 customer 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 (10) for definitions).*

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

$$\forall \mathscr{S} \subset \mathcal{S}_k : U_{\breve{M}}(\mathscr{S}) \subset \mathcal{C}_k \quad \text{and} \quad \mu_{\mathscr{S}} > \Lambda_{U_{\breve{M}}(\mathscr{S})}.$$

Further, since $U_M(\mathscr{S}) \subseteq U_{\breve{M}}(\mathscr{S})$, the positive slack condition also holds for $U_M(\mathscr{S})$. ($U_M(\mathscr{S})$ is the subset of customer classes that can only be served by servers in $\mathscr{S}$ under the menu M.)

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

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

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

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

$$U_M(\mathcal{S}_k) = \mathcal{C}_k \cup \mathcal{C}'_k.$$

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

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

Proof of Lemma 8 There are two differences between the setup in our paper and in Afèche et al. [3]: first, the constants γ_i for the approach to heavy traffic are allowed to be arbitrary, while in Afèche et al. [3] the authors impose $\gamma_i = \Lambda_i$. Second, our setup has customer classes with $\Lambda_i = 0$ and hence CRP components which consist of a single customer class and no servers. Despite these, the proofs for parts (i) and (ii) are identical to the proofs of parts (i) and (ii) of [3, Lemma3].

Part (iii) of [3, Lemma 3] states that $U_M(\mathcal{S}_k) = \mathcal{C}_k$, which in our setup should be interpreted as

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

In addition, a server-less CRP component $\mathbb{C}_\kappa = (\{i\}, \emptyset)$ consisting of a single customer class i is part of the set of customer classes uniquely served by the set $U_M(\mathcal{S}_k)$ 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 $\Delta(\mathcal{S})$ and part (iii):

$$\Delta(\mathcal{S}_k) = \mu_{\mathcal{S}_k} - \lambda_{U_M(\mathcal{S}_k)} = \sum_{\ell=1}^{k} \mu_{\mathcal{S}_\ell} - \sum_{\ell=1}^{k} \sum_{\kappa \in \mathsf{comps}(\sigma,\ell)}$$

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

□

Proof of Lemma 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 σ_t, we have that

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

That is, the customer 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 customer classes in $\mathscr{C}$ using the scaled mean waiting time of a $M/M/1$ queue:

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

Further, from Lemma 8 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 Lemma 2, $\sum_{\ell=1}^{\kappa} \widetilde{\gamma}_{\mathsf{comps}(\sigma_t, \ell)} \leq 0$, then the right-hand side of (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 (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 $\kappa \in [K']$. □

Appendix B Section 4 proofs

The following lemma will be useful in proving Proposition 2 and other results.

Lemma 9 *Let M be an admissible menu with (Λ, μ, γ), and let $\breve{M}$ be the menu given by the residual matching of M. Let $\{\mathbb{C}_1, \ldots, \mathbb{C}_{K'}, \mathbb{C}_{K'+1}, \ldots, \mathbb{C}_K\}$ be the CRP decomposition of M, and hence also of $\breve{M}$. Let σ be a permutation of CRP components with $\widetilde{\Lambda}_k > 0$, (not necessarily a topological order), and let $\mathscr{S} \subseteq [m]$ be a set of servers such that*

$$\mathscr{S} = \bigcup_{i=1}^{k} \mathcal{S}_{\sigma(i)} \tag{A1}$$

for some $k \in [K']$. Then the following statement holds:

(i) *For the menu* $\breve{M}$,

$$\lim_{\epsilon \to 0} \Delta_{\breve{M}}(\mathscr{S}) = \mu_{\mathscr{S}} - \Lambda_{U_{\breve{M}}(\mathscr{S})} = 0.$$

For subsets of servers $\mathscr{S} \subseteq [m]$ *not satisfying Eq.* A1 *for any permutation of CRP components* σ *or integer* k*, the following statement holds:*

(ii) *For the menu* $\breve{M}$

$$\lim_{\epsilon \to 0} \Delta_{M}(\mathscr{S}) \geq \lim_{\epsilon \to 0} \Delta_{\breve{M}}(\mathscr{S}) = \mu_{\mathscr{S}} - \Lambda_{U_{\breve{M}}(\mathscr{S})} > 0.$$

Proof of Lemma 9 Part (i) can be proved as follows. Recall that $\mathcal{C}_k$ is the set of customer classes in $\mathbb{C}_k$, and $\mathcal{S}_k$ is the set of servers. Due to the construction of $\breve{M}$, servers in $\mathcal{S}_k$ are only compatible with customers in $\mathcal{C}_k$. So it suffices to show that $\mu_{\mathcal{S}_k} - \Lambda_{U_{\breve{M}}(\mathcal{S}_k)} = 0$ for all $k \in [K']$.

From Lemma 8 we know that $\Lambda_{\mathcal{C}_k} = \mu_{\mathcal{S}_k}$, so the result will hold if $U_{\breve{M}}(\mathcal{S}_k) = \mathcal{C}_k$. From the construction of $\breve{M}$, we know that $U_{\breve{M}}(\mathcal{S}_k) \subseteq \mathcal{C}$, since only servers in $\mathcal{S}_k$ can serve customers in $\mathcal{C}_k$. Additionally, since no customer class in $\mathcal{C}_k$ can be served by a server not in $\mathcal{S}_k$, every customer on $\mathcal{C}_k$ is also in $U_{\breve{M}}(\mathcal{S}_k)$. Thus $U_{\breve{M}}(\mathcal{S}_k) = \mathcal{C}_k$ and part (i) holds.

Part (ii) can be proved following the arguments in [3, Lemma4]. □

Proof of Proposition 2 To show M is admissible for $(\Lambda - \epsilon\Lambda, \mu)$, we must show that

$$0 < \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_M(\mathscr{S})} \Lambda_i + \epsilon \sum_{i \in U_M(\mathscr{S})} \Lambda_i = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m]. \quad \text{(A2)}$$

The admissibility of M for (Λ, μ, γ) implies that $\sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_M(\mathscr{S})} \Lambda_i \geq 0$ for all $\mathscr{S} \subseteq [m]$. For any $\mathscr{S} \subseteq [m]$ such that $\sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_M(\mathscr{S})} \Lambda_i > 0$, Eq. A2 holds regardless of the ϵ terms. In the case that $\sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_M(\mathscr{S})} \Lambda_i = 0$, then

$$\sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_M(\mathscr{S})} \Lambda_i + \epsilon \sum_{i \in U_M(\mathscr{S})} \Lambda_i = \epsilon \sum_{i \in U_M(\mathscr{S})} \Lambda_i = \epsilon \sum_{j \in \mathscr{S}} \mu_j.$$

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

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

$$0 < \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\breve{M}}(\mathscr{S})} \Lambda_i + \epsilon \sum_{i \in U_{\breve{M}}(\mathscr{S})} \Lambda_i = \Omega(\epsilon) \qquad \text{for all } \mathscr{S} \subseteq [m]. \quad \text{(A3)}$$

There are two cases to consider. In the first case, $\mathscr{S} \subseteq [m]$ satisfies Eq. A1 for some permutation of CRP components σ. In this case, Lemma 9 part (i) applies, and $\sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U_{\breve{M}}(\mathscr{S})} \Lambda_i = 0$. In the second case, $\mathscr{S} \subseteq [m]$ does not satisfy Eq. A1 for any permutation of CRP components. In this case, Lemma 9 part (ii) applies,

and $\sum_{j\in\mathscr{S}}\mu_j - \sum_{i\in U_{\check{M}}(\mathscr{S})}\Lambda_i > 0$. In both cases, Eq. A3 holds following similar reasoning as in the first part of the proposition. □

Appendix C Section 5 proofs

Proof of Proposition 3 Note from (13) that

$$w_{\sigma,k} = \sum_{\kappa=\mathsf{comps}^{-1}(\sigma,k)}^{K'} \frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}} = \frac{1}{\langle\gamma\rangle} + \sum_{\kappa=\mathsf{comps}^{-1}(\sigma,k)}^{K'-1} \frac{1}{\sum_{\ell=1}^{\kappa}\widetilde{\gamma}_{\mathsf{comps}(\sigma,\ell)}}$$

Here we set the empty sum to be zero (i.e. the case in which $K'-1<\kappa$). From Lemma 2 we know that the last summation is non-negative. This combined with (14) tells us that $W_{\mathbb{C}_k} \geq 1/\langle\gamma\rangle$.

Let us now prove the second part of the proposition From the previous discussion, it follows that the requirement $\widehat{W}_{\mathbb{C}_{\hat{\kappa}}} = 1/\langle\gamma\rangle$ can only be satisfied if $w_{\sigma,\hat{\kappa}} = 1/\langle\gamma\rangle$ for all permutations σ associated a topological order. But this can only happen if $\sigma^{-1}(\hat{\kappa}) = K$ for all topological orders $\sigma \in \mathcal{T}(\mathcal{D}, K')$. Since a CRP component $\mathbb{C}_{k_1}$ must come later in every topological order than a component $\mathbb{C}_{k_2}$ that it has a directed arc to in the DAG, this means that a component $\mathbb{C}_\kappa$ is last in every topological order σ if and only if there is a directed path from $\mathbb{C}_\kappa$ to all other CRP components in the DAG, proving the result. This condition is trivially satisfied if $K=1$. □

Proof of Proposition 4 Take any slacks γ with $\langle\gamma\rangle > 0$. We will first show that $M \in \mathcal{M}(\Lambda,\gamma,\mu)$. To do this, we need to show that

$$0 < \Delta^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon) \qquad \text{for all } \mathscr{S}\subseteq[m],$$

where $\Delta^{(\epsilon)}(\mathscr{S})$ is as defined in Definition 2.

We define $D(\mathscr{S})$ as

$$D(\mathscr{S}) = \sum_{j\in\mathscr{S}}\mu_j - \sum_{i\in U(\mathscr{S})}\Lambda_i$$

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

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

From the definition of M we know that $D(\mathscr{S})>0$ for all $\mathscr{S}\subsetneq[m]$, implying that $0<\Delta^{(\epsilon)}(\mathscr{S})=\Omega(\epsilon)$ for all $\mathscr{S}\subsetneq[m]$. For the case of $\mathscr{S}=[m]$, since $\langle\lambda\rangle=\langle\mu\rangle$, and $\langle\gamma\rangle>0$,

$$0 < \Delta^{(\epsilon)}(\mathscr{S}) = \epsilon\langle\gamma\rangle + o(\epsilon) = \Omega(\epsilon)$$

as required.

What remains to be shown is that M induces a single CRP component. This follows from part (i) of Lemma 8, which states that within a CRP component $\widetilde{\Lambda}_k := \Lambda_{\mathcal{C}_k} = \mu_{\mathcal{S}_k} =: \widetilde{\mu}_k$ (see (10) 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 customer 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 labeled so that $(\sigma(k) = k$ 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 $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_K\}$. 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 customer 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.

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 $\mathcal{F}(0, \Lambda, \breve{M}) \subseteq \mathcal{F}(0, \Lambda, M)$. So all we need to show is that there are no flows in M that are not also in $\breve{M}$.

First note that the servers in $\mathcal{S}_k$ are only compatible with customer classes in $\mathcal{C}_k \cup \mathcal{C}_{k+1}$ for $k \in [K - 1]$, and so all flow into servers from $\mathcal{S}_k$ must come from customers in $\mathcal{C}_k \cup \mathcal{C}_{k+1}$. Similarly, servers in $\mathcal{S}_K$ are only compatible with customers in $\mathcal{C}_K$, and so all flow into servers in $\mathcal{S}_K$ must come from customers in $\mathcal{C}_K$.

From Lemma 8 part (i), we know that $\widetilde{\Lambda}_1 = \widetilde{\mu}_1$. Since customers in $\mathcal{C}_1$ are only compatible with servers in $\mathcal{S}_1$, this means that all of the capacity of servers in $\mathcal{S}_1$ is allocated to customers in $\mathcal{C}_1$, even though they are also compatible with customers in $\mathcal{C}_2$. Therefore there is no flow between servers in $\mathcal{S}_1$ and customers not in $\mathcal{C}_1$. Using similar reasoning, 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 σ. 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 σ_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 σ, 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 $0 < \Delta^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon)$ for all $\mathscr{S} \subseteq [m]$, where

$$\Delta^{(\epsilon)}(\mathscr{S}) := \sum_{j \in \mathscr{S}} \mu_j - \sum_{i \in U(\mathscr{S})} \lambda_i^{(\epsilon)}.$$

First consider the case in which $\mathscr{S}$ does not satisfy Eq. A1 for any permutation of CRP components σ. Then from Lemma 10 part (ii), we know that

$$\mu_S - \Lambda_{U(\mathscr{S})} > 0.$$

This means that $0 < \Delta^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon)$ for all $\mathscr{S} \subseteq [m]$ that is not equal to $\cup_{\ell=1}^{k} \mathcal{S}_\ell$ for some $k \in [K]$. (Note that the proof of Lemma 10 does not rely on this result.)

Now consider $\mathscr{S}$ satisfying Eq. A1 for some permutation of CRP components σ. There are two possibilities. First consider $\sigma(k) = k$ for all $k \in [K]$, i.e. the only topological ordered admitted by the DAG on M. In this case, the arguments from Lemma 8 part (iv) hold, and

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

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 $0 < \Delta^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon)$ as required.

For any other permutation of CRP components σ, arguments made in [3] can be used to show that

$$\mu_S - \Lambda_{U(\mathscr{S})} > 0.$$

This means that $0 < \Delta^{(\epsilon)}(\mathscr{S}) = \Omega(\epsilon)$. Hence M is admissible as claimed.

This also demonstrates why no admissible menu M can admit a topological order σ 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 $\lim_{\epsilon \to 0} \Delta^{(\epsilon)}(\mathscr{S}) \leq 0$ 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 $\lim_{\epsilon \to 0} \Delta^{(\epsilon)}(\mathscr{S}) > 0$. □

Proof of Proposition 5 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 minimizes delay for each topological order).

Therefore, we will be able to minimize 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. □

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 $\sigma(\kappa) \leq \sigma(k)$. This means that in every topological order $\sigma \in \mathcal{T}(\mathcal{D}, K')$, $\sigma(\kappa) > \sigma(k)$. From the definition of the conditional delay $w_{\sigma,k}$ in Eq. 13, 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 6 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 9. As stated in the proposition, we will assume there exists a vector $\widehat{\mathbb{W}} = (\widehat{\mathbb{W}}_1, \ldots, \widehat{\mathbb{W}}_L) \in \mathbb{R}_+^L$ such that

(i) $W_k = \widehat{\mathbb{W}}_\ell$ for all $k \in [K]$ such that $\mathbb{C}_k \in \mathscr{C}_\ell$ for some $\ell \in [L]$,
(ii) $\widehat{\mathbb{W}}_\ell < \widehat{\mathbb{W}}_{\ell+1}$ for $\ell = 1, \ldots, L-1$.

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 $\tilde{\gamma}_k = \widehat{\gamma}_\ell$ 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 Eq. 12 is also constant, independent of σ. 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 $\mathbb{W}_\ell$. One can show from Theorem 1 that

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

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

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

Thus, we require that $\widehat{\gamma}_L$ satisfy

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

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 (A1)

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

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 ∞. □

Appendix D Section 6.1 proofs

Lemma 10 *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

$$\{s_1, \ldots, s_\ell\} = \bigcup_{i=1}^{k} \mathcal{S}_{\sigma(i)}. \tag{A1}$$

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 σ for which (A1) is satisfied.

Proof of Lemma 10 The first part follows from the proof of [3, Lemma4] where it is argued that if the subset $S = \{s_1, \ldots, s_\ell\}$ does not obey the condition mentioned, then

$$\mu_S - \Lambda_{U(\mathscr{S})} > 0,$$

and hence $\lim_{\epsilon \to 0} \frac{\epsilon}{\Delta(S)} = 0$. The second part follows from part (iv) of Lemma 8. □

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.*

(i) *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*

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

(ii) *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* (12) *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.*

Proof of Proposition 7 The proof of part (i) follows exactly the same lines as [3, Proposition 2] and hence we omit it. The calculations for part (ii) 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

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

From (18)

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

By Lemma 10,

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

For some $k \in [K']$, and $m_{k-1} + 1 \le \ell \le m_k - 1$, denote $S = \{s_{m_{k-1}+1}, \ldots, s_\ell\}$. Lemma 8 implies that:

$$\lim_{\epsilon \to 0} \Delta(s_1, \ldots, s_\ell) = \mu_S - \Lambda_{U_{\tilde{M}}(\mathscr{S})} > 0.$$

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

$$\theta_k(\mathbf{s}_k) = \prod_{\ell=1}^{|\mathcal{S}_k|-1} \frac{1}{\mu_{\{s_k(1),\ldots,s_k(\ell)\}} - \Lambda_{U_{\tilde{M}}(s_k(1),\ldots,s_k(\ell))}}. \tag{A2}$$

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 5 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_ℓ for $\ell \in \{0,1,\ldots,K'-1\}$ by

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

Define $j(s,i) = \min\{\ell : i \in U(s_1,\ldots,s_\ell)\}$, and define κ satisfying $m_{\kappa-1}+1 \le j \le m_\kappa$. Then, using Lemma 4, 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 10,

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

The last equality follows because the second term in the preceding expression is 0 by Lemma 10, and each of the terms in the first sum is precisely of the form (A1) in Lemma 10. The Lemma now follows by noting that κ only depends on the CRP component $\mathbb{C}_k$ that customer class i belongs to and therefore so does the last expression, and $\kappa = \mathsf{comps}^{-1}(\sigma,k)$. □

Appendix E Section 6.2 proofs

Lemma 11 *For a given admissible service menu M with limiting arrival rates Λ, service rates μ, and slacks Γ, 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 $\sigma \in \mathcal{T}(\mathcal{D}, K')$, $\lim_{\epsilon \to 0} \sum_{x \in P(s,m)} \pi(x) q^0_{ij}(x) = 0$.*

Proof of Lemma 11 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

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

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

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

□

Proof of Lemma 6 Recall from Definition 10 that since the permutation of servers s is induced by the topological order σ, 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

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

Also denote for $s_k \in \Sigma_k$

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

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 Eq. 12 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

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

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

Proof of Lemma 7 From Proposition 7, we know that

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

where $\theta_k(s_k)$ is given by Eq. A2.

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

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

This means that

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

$$= \mathcal{B}'_M \sum_{\sigma\in\mathcal{T}(\mathcal{D},K')} \left[\mathbb{Q}(\sigma) \sum_{\substack{s=(\mathbf{s}_{\sigma(1)}||\mathbf{s}_{\sigma(2)}||\cdots||\mathbf{s}_k||\cdots||\mathbf{s}_{\sigma(K')}) \\ \{\mathbf{s}_k\in\Sigma_{\mathcal{S}_k}\}_{\kappa\in[K']}}} \prod_{k=1}^{K'}\theta_k(\mathbf{s}_k)\right] \tag{A4}$$

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

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

Recall from Sect. 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 Eq. A5, we have that

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

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

References

1. Adan, I., Weiss, G.: Exact FCFS matching rates for two infinite multitytpe sequences. Oper. Res. **60**(2), 475–489 (2012)
2. Adan, I., Weiss, G.: A skill based parallel service system under FCFS-ALIS—steady state, overloads and abandonments. Stoch. Syst. **4**(1), 250–299 (2014)
3. Afèche, P., Caldentey, R., Gupta, V.: On the optimal design of a bipartite matching queueing system. Oper. Res. **70**(1), 363–401 (2022)
4. Atar, R.: A diffusion regime with nondegenerate slowdown. Oper. Res. **60**(2), 490–500 (2012)
5. Bell, S.L., Williams, R.J.: Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: asymptotic optimality of a threshold policy. Ann. Appl. Probab. **11**(3), 608–649 (2001)

6. Bušić, A., Gupta, V., Mairesse, J.: Stability of the bipartite matching model. Adv. Appl. Probab. **45**(2), 351–378 (2013)
7. Caldentey, R., Kaplan E.: A heavy traffic approximation for queues with restricted customer-service matchings. Unpublished manuscript (2002)
8. Caldentey, R., Kaplan, E., Weiss, G.: FCFS infinite bipartite matching of severs and customers. Adv. Appl. Probab. **41**(3), 695–730 (2009)
9. Caldentey, R. Gupta, V., Hillas, L.A.: Designing service menus for bipartite queueing systems. Oper. Res. (forthcoming) (2023)
10. Fazel-Zarandi, M., Kaplan, E.: Approximating the first-come, first-served stochastic matching model with ohm's law. Oper. Res. **6**, 1423–1432 (2018)
11. Gardner, K., Righter, R.: Product forms for FCFS queueing models with arbitrary server-job compatibilities: an overview. Queueing Syst. **96**, 3–51 (2020)
12. Green, L.: A queueing system with general-use and limited-use servers. Oper. Res. **33**(1), 168–185 (1985)
13. Halfin, S., Whitt, W.: Heavy-traffic limits for queues with many exponential servers. Oper. Res. **29**(3), 567–588 (1981)
14. Harrison J.M.: Brownian models of queueing networks with heterogeneous customer populations. In: Stochastic Differential Systems, Stochastic Control Theory and Applications, pp. 147–186. Springer (1988)
15. Harrison, J.M., Lopez, M.J.: Heavy traffic resource pooling in parallel-server systems. Queueing Syst. Theory Appl. **33**, 339–368 (1999)
16. Hurtado Lange, D.A., Maguluri, S.T.: Heavy-traffic analysis of queueing systems with no complete resource pooling. Math. Oper. Res. **47**(4), 3129–3155 (2022)
17. Kaplan E.: Managing demand for public housing. Technical report. ORC Technical Report # 183, MIT (1984)
18. Kaplan, E.: A public housing queue with reneging and task-specific servers. Decis. Sci. **19**, 383–391 (1988)
19. Kingman, J.F.: On queues in heavy traffic. J. R. Stat. Soc. Ser. B (Methodol.) **24**(2), 383–392 (1962)
20. Kushner, H.J., Chen, Y.: Optimal control of assignment of jobs to processors under heavy traffic. Stoch. Int. J. Probab. Stoch. Process. **68**(3–4), 177–228 (2000)
21. Laws, C.: Resource pooling in queueing networks with dynamic routing. Adv. Appl. Probab. **24**, 699–726 (1992)
22. Mairesse, J., Moyal, P.: Stability of the stochastic matching model. J. Appl. Probab. **53**, 1064–1077 (2017)
23. Mandelbaum, A., Stolyar, A.L.: Scheduling flexible servers with convex delay costs: heavy-traffic optimality of the generalized cμ-rule. Oper. Res. **52**(6), 836–855 (2004)
24. Moyal, P., Perry, O.: On the instability of matching queues. Ann. Appl. Probab. **27**(6), 3385–3434 (2017)
25. Pesic, V., Williams, R.: 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. Stoch. Syst. **6**(1), 26–89 (2016)
26. Schwartz, B.: Queueing models with lane selection: a new class of problems. Oper. Res. **22**(2), 331–339 (2004)
27. Shah, V., de Veciana, G.: Asymptotic independence of servers' activity in queueing systems with limited resource pooling. Queueing Syst. **83**(1–2), 13–28 (2016)
28. Talreja, R., Whitt, W.: Fluid models for overloaded multi-class many-service queueing systems with FCFS routing. Manag. Sci. **54**(1), 1513–1527 (2008)
29. Varma, S.M., Maguluri, S.T.: Transportation polytope and its applications in parallel server systems. https://arxiv.org/abs/2108.13167 (2021)
30. Whitt, W.: Heavy traffic limit theorems for queues: a survey. In: Mathematical Methods in Queueing Theory, pp. 307–350. Springer (1974)
31. Williams, R.J.: On dynamic scheduling of a parallel server system with complete resource pooling. Fields Inst. Commun. **28**(49–71), 5–1 (2000)