informs
http://pubsonline.informs.org/journal/mnsc/

MANAGEMENT SCIENCE
Vol. 65, No. 1, January 2019, pp. 152–187
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Robust Multiclass Queuing Theory for Wait Time Estimation in Resource Allocation Systems

Chaithanya Bandi,[a] **Nikolaos Trichakis,**[b] **Phebe Vayanos**[c]

[a] Kellogg School of Management, Northwestern University, Evanston, Illinois 60208; [b] MIT Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; [c] Departments of Industrial and Systems Engineering and Computer Science and Center for Artificial Intelligence in Society, Viterbi School of Engineering, University of Southern California, Los Angeles, California 90089

Contact: c-bandi@kellogg.northwestern.edu (CB); ntrichakis@mit.edu, http://orcid.org/0000-0002-8324-9148 (NT); phebe.vayanos@usc.edu (PV)

Received: May 3, 2016
Revised: February 28, 2017; June 12, 2017
Accepted: August 16, 2017
Published Online in Articles in Advance: May 14, 2018

https://doi.org/10.1287/mnsc.2017.2948

Copyright: © 2018 INFORMS

Abstract. In this paper, we study systems that allocate different types of scarce resources to heterogeneous allocatees based on predetermined priority rules—the U.S. deceased-donor kidney allocation system or the public housing program. We tackle the problem of estimating the wait time of an allocatee who possesses incomplete system information with regard, for example, to his relative priority, other allocatees' preferences, and resource availability. We model such systems as multiclass, multiserver queuing systems that are potentially unstable or in transient regime. We propose a novel robust optimization solution methodology that builds on the assignment problem. For first-come, first-served systems, our approach yields a mixed-integer programming formulation. For the important case where there is a hierarchy in the resource types, we strengthen our formulation through a drastic variable reduction and also propose a highly scalable heuristic, involving only the solution of a convex optimization problem (usually a second-order cone problem). We back the heuristic with an approximation guarantee that becomes tighter for larger problem sizes. We illustrate the generalizability of our approach by studying systems that operate under different priority rules, such as class priority. Numerical studies demonstrate that our approach outperforms simulation. We showcase how our methodology can be applied to assist patients in the U.S. deceased-donor kidney waitlist. We calibrate our model using historical data to estimate patients' wait times based on their kidney quality preferences, blood type, location, and rank in the waitlist.

History: Accepted by Yinyu Ye, optimization.

Keywords: queuing theory • robust optimization • resource allocation • healthcare

1. Introduction

In this paper, we deal with the problem of estimating wait times in systems that allocate scarce resources of different types according to some predetermined priority rule, such as first-come, first-served (FCFS). Allocatees are heterogeneous, differing in their preferences over resource types, and possess *incomplete system information* with regard to their relative priority, other allocatees' preferences, and/or resource availability. We take the perspective of an individual allocatee and tackle the estimation problem of his wait time until he is allocated his preferred resources, based on his available information. Technically, this corresponds to a wait time estimation problem for a particular customer in a multiclass, multiserver (MCMS) queuing system for which primitive information about queue populations, customer arrivals, and/or service times is limited. We argue that wait time estimation in such a context is highly relevant to practical problems and that it requires development of a new methodological framework.

A concrete motivation for our research is the plight of patients suffering from end-stage renal disease, which is terminal, and for which only two treatment options, maintenance dialysis and kidney transplantation, are available. The significant and growing number of patients seeking a kidney transplant in the United States (currently 100,434, 110% higher than 15 years ago) register on a national waitlist. Organs procured from donors are offered to blood-type-compatible wait-listed patients according to a national allocation policy that closely resembles FCFS. In the face of often long and variable wait times, accurate estimates of remaining wait time would be valuable to patients for a number of reasons. One relates to the choice of appropriate treatment protocols, since the timing of initiation and subsequent management of dialysis therapy both rely heavily on estimates of wait time (Lee et al. 2008). The decision of whether to accept or reject an offered kidney (e.g., one of marginal quality from an old donor) also relies heavily on estimates of wait time until the next offer, in particular, of a kidney of better quality (Zenios 2005). Apart from informing the aforementioned decisions that could critically impact survivability, accurate wait time estimates can also help improve patient quality of life. For example, reducing uncertainty around wait times could mitigate patient anxiety and facilitate planning of life activities around dialysis treatment, which roughly entails 12 hours of visits weekly to a dialysis center.

To the best of our knowledge, no tools exist for estimating wait times until offer of a kidney, never mind of a kidney of a particular quality.[1] Our private communications with a number of healthcare providers and physicians at major transplant centers in the New England area corroborated this state of affairs, and attested to the hardship that faces these parties in advising patients about likely wait times to offer of a kidney of acceptable quality. This is hardly surprising considering the following challenges to deriving wait time estimates in this context. First, wait times critically depend on the acceptance propensity of higher-ranked patients, whose preferences with respect to acceptable kidney qualities are unobservable. Second, the allocation system is neither stationary nor stable, with the number of wait-listed patients continually growing, already far exceeding the supply of organs.

The challenges to estimating wait times are not unique to the kidney allocation system (KAS) but are rather usually encountered in systems that allocate scarce goods, especially public ones. Another such system is the U.S. Public Housing Program (PHP), which provides affordable rental housing to low-income families and individuals. The PHP operates in a similar fashion as the KAS: eligible applicants register on waitlists and are offered housing options (that differ in the number of bedrooms, wheelchair accessibility, etc.) as they become available. Specifically, the PHP operates in an FCFS fashion, although some programs accommodate local variations (see Section 6). Wait time estimates are valuable to applicants, because access to affordable housing can have important financial life-planning consequences. Unfortunately, these estimates are equally hard to derive for many of the same reasons as within the KAS—i.e., incomplete information and transient/unstable system behavior (see Section 3 for details). Indeed, all of the housing offices we surveyed in the New England area refrain from providing any but crude, wide-ranging estimates (the Boston housing office, for example, quotes wait times ranging from 10 weeks to more than five years).

Our research objective is to estimate wait times of allocatees based on their own preferences and characteristics, and the limited information they might possess. That is, in this paper, we take the perspective of an individual allocatee, for whom we attempt to derive wait time estimates, taking the underlying resource allocation mechanism as given. For example, we aim to estimate wait times for patients in the KAS based on their own kidney-quality preferences, current rank on the waitlist, and blood type. We model the allocation system as an MCMS queuing system serving customers (the allocatees) in which server multiplicity captures resource heterogeneity (for example, kidneys of different quality) and class multiplicity captures customer heterogeneity (for example, with respect to acceptable kidney qualities). In this setting, our research question deals with the problem of estimating the wait time of a particular customer in a given class based on limited information about queue populations, customer arrival times, and service times.

The large body of work in the queuing literature that deals with MCMS systems is not well suited to our research question posed within systems plagued by incomplete information and/or characterized by transient, potentially unstable behavior—i.e., queuing systems that accurately capture intricacies often encountered in resource allocation in practice (see the discussions in Sections 1.1 and 2). We consequently utilize robust optimization tools known to cope well with information incompleteness and to support the derivation of tractable optimization formulations.

In particular, we develop a new methodological framework for analyzing wait times of customers served by potentially nonstationary or unstable MCMS systems that operate according to predetermined priority rules under incomplete information. Our framework does not postulate probability distributions for the uncertain parameters and instead models stochasticity by means of optimization variables that lie in uncertainty sets, which encompass all available limited information, in the spirit of recent robust queuing theory. We quantify wait times through their worst-case values, which we refer to as robust wait times.

The key challenge in analyzing MCMS systems—namely, to capture the customer–server allocation dynamics implied by a specific priority rule—as we discuss later makes our analysis fundamentally different from existing approaches in robust queuing theory. We address it by introducing a modeling formulation that leverages assignment variables and affords the flexibility of dealing with various priority rules that can be modeled as constraints on the assignment variables. We base our analysis on MCMS FCFS systems, motivated by the KAS and PHP. We illustrate later how our approach can accommodate alternative priority rules. Our formulations, by building on top of assignment problems, exhibit enhanced computational performance. Although the use of assignment variables is motivated by work in the stochastic server allocation and job scheduling literature, the linkage between the robust queuing system and the assignment problem is novel—see our discussion in Section 1.1.

Using our methodological framework, we first derive a mixed-integer programming (MIP) formulation to compute robust wait times in a general MCMS system. We then focus on a subclass of MCMS systems for which there is a hierarchy of resource types. This important subclass, termed hierarchical MCMS (HMCMS), subsumes many practical systems, including the KAS. We leverage the structure of HMCMS systems to strengthen our general MIP formulation through a drastic variable and constraint reduction.

We further develop a heuristic approach to compute approximate robust wait times in HMCMS systems that involves solving only a convex optimization problem (usually a second-order cone program) with a small number of variables. Critically, we derive an approximation guarantee to back our heuristic that becomes tighter as the problem size increases. We demonstrate the performance of our formulations in terms of accuracy and solution times by conducting extensive numerical studies using simulated data for realistic problem sizes.

We put our methodology into practice in a case study of the KAS. Using highly detailed historical data on wait-listed patients and donated organ offers, we calibrate our model to predict wait times based on patients' wait-list rank and blood type.

We subsequently demonstrate how our methodology can be applied to systems that prioritize customers based on priority rules other than FCFS (see Section 6). In particular, we extend our MIP formulation, heuristic approach, and its approximation guarantee to systems in which priority is driven by customer class.

Our work contributes to the following literature streams. First, it builds on and extends nascent robust queuing theory in a significant way by capturing multiple customer classes. This additional modeling component enables the incorporation of customer heterogeneity. Because, from a technical perspective, this relies on introducing customer allocation dynamics to servers, existing robust queuing theory tools are of little use. We show how, by capturing these dynamics via a novel assignment approach, moderately sized MIP formulations and efficient heuristics that afford a priori error bounds can be derived. Second, our work contributes to the broader queuing literature by providing an estimation procedure for wait times in MCMS systems that are potentially unstable and/or in a transient regime that is tractable and accurate under incomplete information. Third, the present work adds to the operations research literature that deals with organ allocation by developing the first method for estimating wait times in the KAS.

1.1. Literature Review

1.1.1. Robust Queuing Theory. This nascent literature stream deals with queuing systems under uncertainty in arrival and service times. Xie et al. (2011) use an approach based on the Stochastic Network Calculus framework to propose bounds on the delays in Internet networks in transient regime. Bandi et al. (2015, 2018) model networks of single-class queues using a robust optimization approach via uncertainty sets and obtain bounds on the waiting times using a worst-case-analysis approach. These papers deal with single-class, homogeneous customers, which allows them to build their analysis using the standard Lindley recursion or extensions thereof. Our work is inspired by the use of robust optimization for queuing systems analysis. However, our dealing with customer heterogeneity introduces highly nonlinear dynamics with regard to customer–server routing according to priority rules. These dynamics invalidate the Lindley recursion and consequently the techniques presented in the aforementioned papers.

1.1.2. Multiclass Multiserver Queuing Theory Under Transient Regime. MCMS queuing systems have been a major topic of study given their varied applications. The vast majority of papers in this stream focus on optimal control or stability analysis. Optimal control deals with the derivation of priority rules that optimize certain performance metrics such as throughput, delays, etc.; see e.g., Harrison and Van Mieghem (1997), Jiang and Walrand (2010), Plambeck and Ward (2006). Stability analysis examines conditions and priority rules under which queuing systems are stable; related findings are clearly and elegantly summarized in the survey paper by Bramson (2008). A subclass of MCMS systems that is closer to the ones we consider in this paper is that of parallel-server networks for which Bell and Williams (2001), Harrison and López (1999), and Mandelbaum and Stolyar (2004) again address optimal control and stability issues. In contrast, we deal with systems that (a) operate under predetermined priority rules and (b) are inherently unstable and in transient regime, such as the KAS and the PHP.

Transient analysis of queuing systems began with the analysis of $M/M/1$ queues, for which Karlin and McGregor (1958) showed that it involved an infinite sum of Bessel functions. The analysis was further extended (Abate and Whitt 1987, 1988, 1998; Choudhury et al. 1994; Choudhury and Whitt 1995) to obtain additional insights on the queue length process. In view of the insurmountable tractability challenges even for stable Markovian queues (see, e.g., the discussion in Gross et al. 2008, Heyman and Sobel 2003, Odoni and Roth 1983, and Keilson 1979), several approximation techniques have been proposed, such as the ones by Grassmann (1977, 1980), Kotiah (1978), Moore (1975), Rider (1976), Rothkopf and Oren (1979), and others. All such approaches we are aware of have focused on developing numerical techniques for single-class queues and queuing networks and do not generalize to multiclass queuing systems of the type we study in this paper.

To the best of our knowledge, all papers in this literature stream consider primitive information regarding system dynamics, arrivals, and service durations to be known and specified using distributions. We deal with problems where (pieces of) such information is (are) unavailable.

1.1.3. Optimization Approaches in Multiserver Queuing Systems. A growing stream of research proposes to employ linear and integer optimization for queuing and scheduling problems. Gurvich et al. (2010) consider the problem of jointly optimizing staffing levels and priority rules in a queuing system with uncertain arrivals. To optimize over the priority rule, they treat the number of jobs assigned to each server as optimization variables. Similarly, integer optimization variables are routinely employed in scheduling problems to determine a schedule (or job-to-server assignment) that optimizes a certain objective; see, e.g., Pinedo (1995) or the survey by Queyranne and Schulz (1994). More recently, Deng and Shen (2016) use an assignment-style formulation to derive optimal appointment schedules. Although our assignment-style formulations are motivated by the referenced work here, our work highlights the linkage between the assignment problem and robust queuing system analysis. Furthermore, note that in all referenced work, the job-to-server assignment variables are used to determine an optimal priority rule. Our work differs in that the assignment variables are used to describe the system's evolution under a predetermined priority rule. Consequently, appropriate constraints need to be devised so that feasible assignments respect each given priority rule. From this standpoint, our work mimics Bodur and Luedtke (2017), where the authors use job-to-server assignment variables to capture dynamics under the shadow-tandem priority rule. In contrast, we study a robust queuing setting and focus on FCFS and class priority rules.

1.1.4. Model-Based Organ Allocation. This literature comprises two streams. Papers in the first stream take the perspective of policy makers and devise organ allocation policies that would improve on the status quo. Zenios (2005) provides an excellent survey of earlier work in this stream, whereas more recent papers include Akan et al. (2012), Bertsimas et al. (2013), Kong et al. (2010), and Su and Zenios (2006). Our approach is very different as we consider the U.S. national allocation policy in place and estimate patient wait times.

Papers in the second stream take the patients' perspective and study the accept/reject decision that they face when offered an organ, by modeling it as an optimal stopping problem in an MDP framework. The key insight from these papers is that patients follow threshold-type policies—i.e., each patient has a threshold on organ quality and accepts (rejects) organs if they are above (below) this threshold. See again the survey by Zenios (2005) for earlier papers, and Alagoz et al. (2007), Sandıkçi et al. (2008), and Sandıkçi et al. (2013) for recent work. Our work takes a different angle: we borrow the key insight of these papers—that is, we take as given that the patients' accept/reject behavior is threshold-type and focus on characterizing the time until the next offer. This angle is in some sense complementary to the existing papers, which take as given a characterization of the time until the next offer and focus on the accept/reject decision problem.

1.2. Notation

We denote sets (resp. random variables) using uppercase blackboard bold (resp. uppercase script) typeface style. Superscripts affixed to vectors are used for element indexing—e.g., if $x_{ij} \in \mathbb{R}^k$, then x_{ij}^{ℓ} is its ℓth element. We denote the indicator function with $\mathcal{I}(\cdot)$. Finally, e is the vector of all ones, and e_i is the vector with its ith element equal to one and all other elements equal to zero.

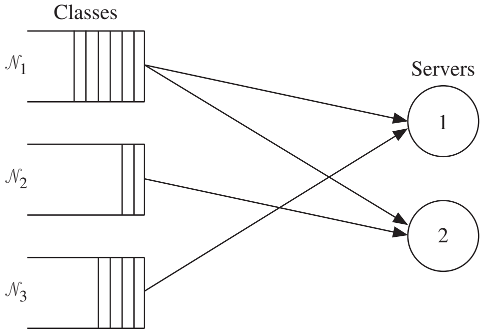
2. Model

We begin by developing a queuing model that can be used to analyze wait times in first-come first-served (FCFS) resource allocation systems—e.g., the kidney or public housing allocation systems discussed in the introduction. To obtain a general-purpose model that can be widely applicable, we omit capturing particularities of specific applications. We next present the model, followed by a discussion of how it can be applied to tackle our research questions.

Consider a multiclass, multiserver (MCMS) queuing system where a set of M distinct servers, indexed by $j = 1, \ldots, M$, serve K customer classes, similarly indexed by $i = 1, \ldots, K$. Associated with the ith customer class, there is an infinitely sized queue that is populated by all customers of that class, which we shall refer to as *i-customers*. Customers of each class can only be served by a fixed subset of servers. Let $\mathbb{S}(i) \subset \{1, \ldots, M\}$ be the (nonempty) set of servers *eligible* to serve i-customers. Correspondingly, let $\mathbb{Q}(j) \subset \{1, \ldots, K\}$ be the set of queues or customer classes for which the jth server is eligible. Figure 1 provides an illustrative example.

At time $t = 0$, there is a (random) number $\mathcal{N}$ of customers waiting for service in the system, with $\mathcal{N}_i$ of them being i-customers. We index customers by $\nu = 1, \ldots, \mathcal{N}$ so that $\{1, \ldots, \mathcal{N}_1\}$ are 1-customers, $\{\mathcal{N}_1 + 1, \ldots, \mathcal{N}_1 + \mathcal{N}_2\}$ are 2-customers, etc. Customers are served according to FCFS. Let σ be a permutation of $\{1, \ldots, \mathcal{N}\}$ that produces the order in which the $\mathcal{N}$ customers arrived. In particular, $\sigma(\nu)$ is the order in which the νth customer arrived—and thus his service priority as well. The system is closed after $t = 0$—i.e., no more

Figure 1. Illustration of a Multiclass Multiserver Queuing System with $M = 2$ Servers and $K = 3$ Classes (Queues), for Which $\mathbb{S}(1) = \{1, 2\}$, $\mathbb{S}(2) = \{2\}$, $\mathbb{S}(3) = \{1\}$, and $\mathbb{Q}(1) = \{1, 3\}$, $\mathbb{Q}(2) = \{1, 2\}$



customers arrive.[2] Subsequently, to receiving service by any eligible server, customers exit the system.

We assume without loss that all servers are busy at $t = 0$. Service times of the jth server are independent and identically distributed (i.i.d.)—in particular, independent of customer class—and are denoted by $\{\mathcal{X}_j^\ell\}_{\ell\in\mathbb{N}}$. Specifically, after $t = 0$ the jth server becomes available for service for the first time at $t = \mathcal{X}_j^1$, it then begins servicing another customer, becoming available again at $t = \mathcal{X}_j^1 + \mathcal{X}_j^2$, etc. Let μ_j be the average service rate of the jth server and $1/\mu_j$ its average service time. Service times are also assumed independent across servers and independent of queue populations.

Once a server becomes available, it immediately starts servicing the highest-priority customer among the remaining ones for which the server is eligible. To formalize this, let $\mathbb{L}_i(t)$ be the set of i-customers waiting in the ith queue at time t. For example, as per our aforementioned indexing convention, we have that $\mathbb{L}_1(0) = \{1, \ldots, \mathcal{N}_1\}$. Suppose that the jth server becomes available at time t. The server then starts servicing customer $\nu^\star \in \arg\min\{\sigma(\nu)\colon \nu \in \bigcup_{i\in\mathbb{Q}(j)} \mathbb{L}_i(t)\}$. Subsequently, customer $\nu^\star$ leaves the queue $i^\star$ in which he waited—i.e., if $\nu^\star \in \mathbb{L}_{i^\star}(t)$, we have $\mathbb{L}_{i^\star}(t+) = \mathbb{L}_{i^\star}(t)\backslash\{\nu^\star\}$. If there are no customers waiting at time t for which the jth server is eligible—i.e., $\bigcup_{i\in\mathbb{Q}(j)} \mathbb{L}_i(t) = \varnothing$—then the server remains idle.

In this setting, the *clearing time* for the ith queue is defined as the time at which it first empties,

$$\begin{aligned}&\mathcal{W}_i(\mathcal{N}_1, \ldots, \mathcal{N}_K, \sigma, \{\mathcal{X}_1^\ell\}_{\ell\in\mathbb{N}}, \ldots, \{\mathcal{X}_M^\ell\}_{\ell\in\mathbb{N}})\\ &\quad := \inf\{t \geq 0\colon |\mathbb{L}_i(t)| = 0\},\end{aligned}$$

and is a complex function of the state of the system at time $t = 0$, described by the queue populations $\mathcal{N}_1, \ldots, \mathcal{N}_K$, the priority mapping σ, and the service times $\{\mathcal{X}_1^\ell\}_{\ell\in\mathbb{N}}, \ldots, \{\mathcal{X}_M^\ell\}_{\ell\in\mathbb{N}}$.

The focal point of our subsequent analysis is to quantify the clearing times of queues in the model described above. Before presenting the analysis, we illustrate how this will allow us to tackle the main research problem we outlined in the introduction. In particular, consider an FCFS multiclass, multiserver queuing system. The wait time of an existing, particular customer corresponds then to the clearing time of the queue he belongs to in an appropriately specified instance of our model. The statistics of the queue populations, the priority order, and the service times in our model can be calibrated so as to reflect the (partial) characterization of the state of the system that is available.

It is important to note here that we do not require the original queuing system one would want to analyze to be closed. For example, the queuing system underlying kidney allocation in the United States is open and unstable—i.e., patients arrive at a higher rate than kidneys. Since wait times of existing customers in FCFS systems are not affected by future arrivals, however, a closed queuing system model suffices for our purposes.

3. Robust Optimization Framework for Multiclass Multiserver Systems

The analysis of MCMS queuing systems like the one we introduced in the previous section has attracted a lot of attention in the queuing theory literature. While this theory offers a considerable arsenal of analysis tools for such systems, the vast majority of them either (a) address alternative questions to ours or (b) rely on assumptions that would be prohibitive for us to make in our setting.

Specifically, the focal points in the MCMS queuing theory literature have been establishing stability of such systems and/or optimizing over priority or control mechanisms (see Section 1.1). For our purposes, however, the key quantity of interest is clearing or wait times under a predetermined priority rule (e.g., FCFS). Among the studies closer to ours that quantify wait times, the majority of them obtain general-purpose averages from a system's perspective. Our focus is on estimating wait times for particular customers in the system who might have already been waiting for some time, based on the unique, limited, and idiosyncratic information they might possess.

Furthermore, studies in the literature quantifying wait times for MCMS systems usually assume that there is complete information, that the system is stable, and that it starts with empty queues. Unfortunately, all of these assumptions are in contrast with the following practical considerations underlying the resource allocation systems we are interested in analyzing:

1. *Incomplete information*: Resource allocation systems of public goods are often plagued by lack of information. For example, patients' preferences pertaining to acceptable organ quality are private information and unobservable in the kidney allocation system (see Section 5). In the public housing allocation system, while

candidates submit their housing preferences at registration, their true preferences might again be unobservable because candidates might not be fully incentivized to reveal them, or because they might change over time. In addition, the construction rate of new housing developments could also be hard to estimate because of limited historical data in developing regions and their dependence on fluctuating socioeconomic factors. From a modeling perspective, this means that probabilistic models of queue populations and/or service distributions might be simply unavailable, or very hard to estimate, to the extent that postulating specific distributional forms might compromise predictive ability.

2. *Instability and transient behavior*: The queuing systems underlying practical resource allocation systems are often unstable, or do not reach steady state during their lifetime, consequently remaining in transient state. For instance, the kidney supply scarcity is well documented, with the number of registered patients waiting for a kidney transplant rising by at least 1,650 every single year since 1995, and on average by 4,750 per year, resulting in ever increasing wait times (Abouna 2008, Horvat et al. 2009). Similarly, wait times in overloaded public housing programs could exceed five years. Such systems, even when they are stable, are unlikely to reach steady state because house availability and new constructions are likely to be heavy tailed and/or time varying because of, for example, fluctuating socioeconomic and policy factors during these long periods (Barabási 2005).

3. *Nonzero initial queues*: The systems we consider do not start from empty, but with a certain queue population in each class already waiting for service. This nonzero initial condition usually leads to analytical intractability when traditional approaches are used for analysis (Kaczynski et al. 2012, Kelton and Law 1985).

All of the reasons outlined in the discussion above motivate us to consider the use of robust optimization tools as an alternative modeling approach to tackling our research questions. In particular, we develop a solution approach inspired by the very recent robust queuing theory (RQT) surveyed in Section 1.1. This theory being limited to single-class queuing systems, we extend the methodology in multiple ways to adequately address MCMS systems—more details on how our work builds on and extends RQT are included in Section 1.1.

3.1. Our Model of Uncertainty

As in RQT, we treat random quantities—e.g., service times—as decision variables in an optimization problem. These variables are constrained to lie in uncertainty sets that reflect fundamental known properties that the original random quantities would satisfy with high probability.

To this end, let x_j^ℓ be the variable corresponding to the ℓth service time of the jth server and n_i the variable corresponding to the number of i-customers in the system—previously denoted by the random variables $\mathcal{X}_j^\ell$ and $\mathcal{N}_i$, respectively. We also let $n := [n_1 \cdots n_K]^\top$ and, for all $j = 1, \dots, M$, let $x_j := [x_j^1 \cdots x_j^{\bar{\ell}_j}]^\top$, where $\bar{\ell}_j$ is an upper bound on the number of customers served by the jth server (we elaborate on how to compute $\bar{\ell}_j$'s later).

In line with RQT and several other recent papers in the robust optimization literature, we constrain the deviations of sums of service times from their means using bounds dictated by the Generalized Central Limit Theorem (GCLT). In particular, we make the following assumption.

Assumption 1. *The service times x_j of the jth server belong to the uncertainty set*

$$\mathbb{X}_j := \left\{ x_j \in \mathbb{R}_+^{\bar{\ell}_j} \colon \sum_{k=1}^{\ell} x_j^k \le \frac{\ell}{\mu_j} + \Gamma_j^{\mathbb{X}} (\ell)^{1/\alpha_j},\ \ell = 1, \dots, \bar{\ell}_j \right\}, \quad j = 1, \dots, M,$$

where $\Gamma_j^{\mathbb{X}} \ge 0$ controls the degree of conservatism, and $\alpha_j \in (1, 2]$ is a heavy tail parameter.

We refer the interested reader to Appendix A and to Bandi and Bertsimas (2012) for a more elaborate motivation and justification of Assumption 1. To streamline our analysis and ease notation, we denote the *completion times* of the jth server (assuming it processes $\bar{\ell}_j$ customers) with $c_j := [c_j^1 \cdots c_j^{\bar{\ell}_j}]^\top$, where $c_j^\ell := \sum_{k=1}^{\ell} x_j^k$, and the uncertainty set they belong to with

$$\mathbb{C}_j := \left\{ c_j \in \mathbb{R}_+^{\bar{\ell}_j} \colon c_j^\ell = \sum_{k=1}^{\ell} x_j^k,\ \ell = 1, \dots, \bar{\ell}_j,\ x_j \in \mathbb{X}_j \right\}, \quad j = 1, \dots, M.$$

While the GCLT-based structure of the uncertainty sets $\mathbb{X}_j$ ($\mathbb{C}_j$) is standard in the robust optimization literature, the structure of an uncertainty set for queue populations could be different and highly context specific. In particular, such a set would need to capture the idiosyncratic information that is available. To preserve generality and tractability, we only assume the following.

Assumption 2. *The queue populations $n \in \mathbb{N}^K$ belong to a bounded polyhedral uncertainty set $\mathbb{P}$.*

The family of linear inequalities is rich enough to capture a vast variety of information pieces that might be available to characterize $\mathbb{P}$. For example, if a patient in the kidney waitlist knows with certainty that there are 10 patients with higher priority ahead of him, the constraint $\sum_{i=1}^{K} n_i = 10$ could capture this information. In Section 5, and for the purposes of our detailed case study on the kidney allocation system, we exemplify how such a set could be constructed in practice.

We do not impose any constraints on the (random) permutation of customers σ that determines service priority. That is, given queue populations $n \in \mathbb{N}^K$, σ could be any permutation of numbers $1, 2, \ldots, \sum_{i=1}^K n_i$. We denote the set of all such possible permutations with $\Sigma(n)$.

3.2. Solution Methodology

We introduce the concept of the *robust wait time* or *robust clearing time* of the ith queue, denoted by W_i, defined as the maximum (worst-case) clearing time subject to the random quantities lying in their uncertainty sets. That is, W_i is the optimal value of the optimization problem

$$\begin{array}{rl} \text{maximize} & \mathcal{W}_i(n_1, \ldots, n_K, \sigma, x_1, \ldots, x_M) \\ \text{subject to} & n \in \mathbb{P} \cap \mathbb{N}^K, \\ & \sigma \in \Sigma(n), \\ & x_j \in \mathbb{X}_j, \quad j = 1, \ldots, M. \end{array} \tag{1}$$

As we shall see, and in line with recent papers in the robust optimization literature, by picking appropriate values for the conservatism parameters $\Gamma_j^{\mathbb{X}}$, one can use W_i as a way to estimate different statistics of the clearing time $\mathcal{W}_i$—e.g., its average, its 95-, 97-percentiles, etc. As a technical remark, we henceforth assume that there exists a population vector n for which the ith class is populated—i.e., $n_i \geq 1$—since otherwise $W_i = 0$.

Before we proceed with the solution of (1), it is important to note that the worst-case estimates this approach can produce are of high practical relevance in the context of service/resource allocation systems. As a matter of fact, in many service systems where demand outstrips supply, managers prefer to provide service guarantees to their customers, instead of average wait time estimates (Aufderheide 1999, Davis et al. 2014, Matas et al. 2015). In healthcare, patients being typically risk averse, worst-case estimates are highly valued and are often used for treatment planning (Elwyn et al. 2001, Entwistle et al. 1998, Vincent and Coulter 2002).

3.2.1. An Assignment Formulation.

Problem (1) is hard to solve, as formalized in our first result.

Proposition 1. *The optimization problem* (1) *is* $\mathcal{NP}$*-hard.*

All proofs are included in Appendix F. Deriving a tractable formulation for (1) is challenging, because there is no analytical expression for $\mathcal{W}_i$. Note that in single-queue settings, Lindley's equations can be used to characterize $\mathcal{W}_i$. For example, the analysis of networks of single-server queues by Bandi et al. (2015, 2018) is based entirely on these equations. In an MCMS setting however, the presence of multiple queues and heterogeneous customers make the system dynamics significantly more complicated. This is because customers waiting in queues need to be routed to servers according to a priority rule (e.g., FCFS). Lindley's equations are insufficient to capture such dynamics and, consequently, an alternative line of attack is needed.

We introduce a novel approach to solve problem (1). The main idea is to model the routing process as an assignment problem, where customers are assigned to servers. Put differently, any permutation σ in problem (1) that determines service/routing priority induces a particular solution to our assignment formulation. The key is that our formulation allows for the reverse as well: by including appropriate constraints on the assignment variables, we ensure that any feasible assignment abides by the FCFS priority discipline under some permutation σ.

Our modeling choice enables us to cast (1) as a mixed-integer optimization problem (MIP). The main decision variables of the MIP are the assignment variables y_{kj}^{ℓ}, which indicate whether the ℓth service that the jth server provides is to a k-customer. Consider the MIP

$$\text{maximize} \quad w_i \tag{2a}$$

subject to

$$\sum_{k \in \mathbb{Q}(j)} y_{kj}^{\ell} \leq 1, \qquad \ell = 1, \ldots, \bar{\ell}_j, j = 1, \ldots, M; \tag{2b}$$

$$\sum_{\substack{\ell = 1, \ldots, \bar{\ell}_j \\ j \in \mathbb{S}(k)}} y_{kj}^{\ell} \leq n_k, \qquad k = 1, \ldots, K; \tag{2c}$$

$$\sum_{k' \in \mathbb{Q}(j)} y_{k'j}^{\ell} \geq f_{kj}^{\ell}, \qquad k \in \mathbb{Q}(j), \ell = 1, \ldots, \bar{\ell}_j, j = 1, \ldots, M; \tag{2d}$$

$$w_k \leq c_j^{\ell} + \bar{\zeta} f_{kj}^{\ell}, \qquad k \in \mathbb{Q}(j), \ell = 1, \ldots, \bar{\ell}_j, = 1, \ldots, M; \tag{2e}$$

$$w_k \geq c_j^{\ell} - \bar{\zeta}(1 - y_{kj}^{\ell}), \qquad k \in \mathbb{Q}(j), \ell = 1, \ldots, \bar{\ell}_j, j = 1, \ldots, M; \tag{2f}$$

$$c_j \in \mathbb{C}_j, \qquad j = 1, \ldots, M; \tag{2g}$$

$$y_{kj}^{\ell}, f_{kj}^{\ell} \in \{0, 1\}, \qquad k \in \mathbb{Q}(j), \ell = 1, \ldots, \bar{\ell}_j, \; j = 1, \ldots, M; \tag{2h}$$

$$(n + \mathrm{e}_i) \in \mathbb{P} \cap \mathbb{N}^K, \tag{2i}$$

with variables $w, n \in \mathbb{R}^K$, $y, f \in \{0, 1\}^{\sum_{j=1}^M |\mathbb{Q}(j)| \bar{\ell}_j}$, $c \in \mathbb{R}^{\sum_{j=1}^M \bar{\ell}_j}$, where $\bar{\zeta}$ is an upper bound on W_i.

Theorem 1. *The optimal value of the MIP* (2) *is equal to* W_i, $i = 1, \ldots, K$.

Apart from the assignment variables y and their associated completion times c, we use the auxiliary variables f to indicate whether a customer class is filled, or has emptied: $f_{kj}^{\ell} = 1$ if at the time the ℓth service of the jth server begins, the k-customers' class is still populated. Constraints (2b) and (2c) are assignment constraints. Constraint (2d) ensures that the jth server will be assigned to customers once it becomes available, unless all classes $\mathbb{Q}(j)$ it is eligible for have emptied. Constraint (2e) can be active only if the kth customer class has emptied, yielding an upper bound on the clearing time of the kth queue. Constraint (2f) provides a nontrivial lower bound on the clearing time of the kth queue whenever an assignment is made to that queue. Constraints (2g) and (2i) ensure that the completion times and queue populations lie in their respective uncertainty sets.[3] Finally, parameters $\bar{\ell}_j$ and $\bar{\zeta}_i$ can be readily calculated as $\bar{\ell}_j = \max\{\sum_{k\in\mathbb{Q}(j)} n_k \colon n \in \mathbb{P} \cap \mathbb{N}^K\}$ and $\bar{\zeta} = \max_j\{\bar{\ell}_j/\mu_j + \Gamma_j^{\mathbb{X}}(\bar{\ell}_j)^{1/\alpha_j}\}$. For more details, see the proof of Theorem 1.

The main appealing features of our methodology are as follows.

1. *Tractability*: The use of assignment variables allow us to capture the complex MCMS dynamics using an MIP formulation, which can be solved via available solvers. More importantly, at the heart of our formulation lies an assignment problem, which is known for its tractability properties. As a matter a fact, the required computational times we recorded in our numerical studies (presented below) demonstrate that instances of practical relevance can be solved in less than few minutes. Furthermore, when dealing with specific applications, one could potentially leverage their structure to strengthen formulation (2), as Section 4 exemplifies.

2. *Generalizability*: While a vast number of MCMS queuing applications follow FCFS and can consequently be analyzed using formulation (2), other priority rules are encountered in practice as well. We argue that our modeling approach is generalizable and offers the potential to capture priority rules other than FCFS. In particular, this would be made possible by imposing appropriate constraints on the assignment variables that would reflect the desired rules. In Section 6, we study a system where a class-priority (CP) rule is followed instead of FCFS, as well as a "hybrid" system where some of the servers follow FCFS and others follow CP. Under CP, the study of open systems becomes relevant and our framework is extended accordingly to capture customer arrivals.

Furthermore, we emphasize that formulation (2) does not rely on the GCLT-based structure of the service time uncertainty sets imposed via Assumption 1. In particular, Theorem 1 applies as long as $\mathbb{X}_j$ are nonempty, bounded polyhedra (see Appendix A).

3. *Robustness*: By relying on a worst-case analysis, our solution approach works very well under a wide range of uncertainty scenarios that could realize in practice and is thus robust to misspecifications of underlying distributions/primitives. For further evidence, we refer the reader to the numerical studies that follow.

3.3. Performance

We performed a wide range of numerical studies to evaluate the accuracy and computational speed of our solution approach in estimating different statistics of clearing/wait times in our model. In particular, we randomly generated multiple instances under different system sizes (varying from $K = M = 5$ to 500), different service distributions (varying from exponential to normal distributions with coefficients of variation between 20% and 40%, to Pareto distributions with parameter α between 1.3 and 1.7) and different average queue populations (varying from 5 to 500).

For all instances, we used our formulation (2) to estimate the average, 95-, 97-, and 99-percentiles of clearing times. We then used a standard simulation approach to approximate these statistics. Assuming that simulation produced the statistics' true values, we measured the average absolute relative error of our estimates as

$$\frac{\displaystyle\sum_{k=1}^{\#\,\text{iterations}} \left| \frac{(\text{our estimate})_k - (\text{simulation estimate})_k}{(\text{simulation estimate})_k} \right|}{\#\,\text{iterations}} \times 100\%.$$

To evaluate the robustness of our estimates to misspecifications of the queue populations' distributions, we also considered cases where the true distributions were different from the ones assumed by the models. In these cases, we used the simulation approach to produce its own estimates under the assumed distribution and measured its errors in a similar fashion as with our approach. Finally, we recorded the required computational times to solve formulation (2) for all generated instances.

We next present only a summary of our results; a detailed description of our experiments and findings is included in Appendix B. Table 1 reports the average absolute relative errors of our approach in estimating different statistics of the clearing times. While these figures are averages across all instances, we note that performance was relatively uniform across different problem sizes and distributions. With regard to computational times, the majority of instances solved within a matter of few seconds, while all instances solved in less than three minutes. In case the true distributions were different from the ones assumed, we found that the relative errors of both our and the simulation approach depended more strongly on the queue population sizes. Table 2 includes the average relative errors we recorded for both approaches for different

Table 1. Average Absolute Relative Errors of Our Clearing Time Statistics' Estimates Across All Instances for Which the True Distributions Were Known

Clearing time statistics	Average	95-percentile	97-percentile	99-percentile
Avg. absolute relative error (%)	6.52	2.64	2.55	3.41

Table 2. Average Absolute Relative Errors of the Simulation's and Our Approach's Estimates for the Average Clearing Time Across All Instances for Which the Assumed Queue Population Distribution Was Different from the True One, for Different Average Queue Populations

Avg. queue population	5	100	500
Simulation's avg. absolute relative error (%)	21	15	12
Our avg. absolute relative error (%)	13	9	7.5

queue population sizes when the true distributions were different from the ones assumed.

Our numerical studies showcase that our methodology provides accurate estimates of clearing time statistics. For practical situations where distributions are unavailable or there is a discrepancy between the assumed and the actual ones, our studies suggest that our methodology would provide far superior estimates compared to simulation, illustrating its usefulness.

4. Hierarchical Service Systems

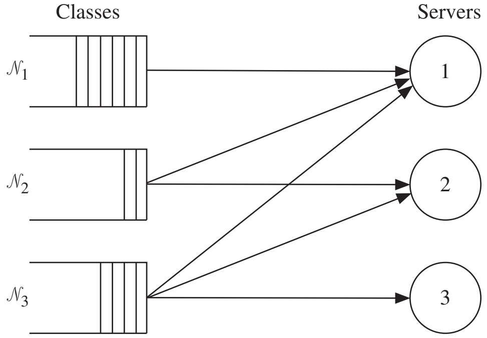
Before applying our methodology to quantify wait times in the U.S. kidney allocation system (KAS), we study an important subclass of MCMS queuing systems that subsumes a vast number of practical applications (including KAS). We leverage structural properties of this subclass to strengthen the MIP formulation (2). We also derive a heuristic to estimate wait times that involves the solution of a scalable convex optimization problem, and back it with a performance guarantee.

In particular, in this section, we study MCMS queuing systems whereby there is a *hierarchy* across the service that the different servers provide, and customers seek service that meets or exceeds a particular rank, or level, in this hierarchy. To make this precise, we assume that the jth server provides the jth highest service level—e.g., server 2 provides the second-highest service level. Correspondingly, i-customers are seeking service level i or higher (e.g., servers 1 and 2 are the only servers eligible for 2-customers); K-customers seek service of any level. From a modeling standpoint, under this hierarchy we have as many customer classes as servers, $K = M$, and the sets $\mathbb{S}$ and $\mathbb{Q}$ have a particular "nested" structure

$$\mathbb{S}(i) = \{1, \ldots, i\}, \quad i = 1, \ldots, K;$$
$$\mathbb{Q}(j) = \{j, \ldots, K\}, \quad j = 1, \ldots, K.$$

We shall refer to such systems as *hierarchical service systems* or *hierarchical* multiclass, multiserver systems (HMCMS). Figure 2 provides an illustrative example for $K = 3$.

Figure 2. Illustration of a Hierarchical Multiclass Multiserver Queuing System with $K = 3$



Notes. Server 1 (3) provides the highest (lowest) service level. Customers in class 1 (2 or 3) seek service at level 1 (2 or 3) and above.

Note that despite being a special case, HMCMS systems arise very frequently in practice, for example, when there are different quality levels of a particular service that is provided. Some concrete examples include (a) kidney allocation, where donated organs have different quality based on donor characteristics;[4] (b) healthcare services, where different technology generations are used with newer ones typically outperforming older ones (e.g., conventional, intensity-modulated or proton radiation therapy services); and (c) transportation services, where different travel classes are offered. In such contexts, it is natural to assume that "customers" who are willing to accept a specific quality of service level will also be willing to accept all higher quality levels; in other words, customer heterogeneity stems only from different quality level thresholds the customers have. This threshold-type customer heterogeneity gives rise to the nested structure of HMCMS systems.

In the remainder of this section and for the case of hierarchical service systems, we leverage their structural properties to strengthen the general formulation (2) so as to compute wait time for service (of any level) in a more efficient manner. We also devise a highly scalable heuristic approach that approximates robust wait times and is backed by a strong approximation guarantee.

4.1. Service Wait Time

An important quantity in the context of hierarchical service systems is the wait time to receive service of any level. This quantity, denoted by $\mathcal{W}_K$ in our framework, corresponds to the wait time a customer will experience if he were to abolish any quality/service level threshold he may have and is a commonly reported metric in hierarchical service systems in practice. For example, the medical reporting website of the government of Alberta, Canada,[5] provides wait time statistics for service of any level for all reported medical procedures (e.g., imaging services, interventions, surgical services) and does not provide a breakdown based on the quality of service or technology used. Similarly, in its Cancer Waiting Times Annual Report,[6] the English National Health Service only reports wait time statistics for cancer services of any level. For example, wait time statistics reported for radiotherapy treatment are agnostic to technology generations. The practical relevance of the quantity $\mathcal{W}_K$ is not surprising: by abstracting away from preferences, it constitutes a baseline measure for wait times as individual preferences could only lead to increased waiting.

Calculating the worst-case $\mathcal{W}_K$ in an HMCMS system remains a hard problem. However, in what follows we leverage its structure to strengthen our formulations.

Proposition 2. *Calculating W_K for HMCMS systems is an $\mathcal{NP}$-hard problem.*

The MIP formulation (2) we proposed to estimate wait times for general MCMS systems involves two sets of key decisions: customer assignment to servers and completion times (captured by variables y and c, respectively). While the former correspond to variables and constraints that appear in assignment problems, which are known to scale well, the latter variables and constraints make formulation (2) deviate from a classical assignment problem, and are thus harder to deal with from a computational standpoint. It turns out that for HMCMS and W_K, completion, or service times can be fixed to their worst-case values in our formulation. This allows us to considerably simplify it, by eliminating the associated variables and constraints.

Before we present more details, we argue that in the computation of W_i, service times need not take their worst-case values in general—even if the system has a hierarchical structure, but $i < K$. We prove this via an example showing that shorter service times could lead to longer wait times.

Example 1. Consider a hierarchical service system with $K = M = 2$ queues, each of which is populated by a single customer—i.e., $\mathbb{P} = \{(1,1)\}$. We are interested in the clearing time W_1 of the first queue—i.e., the waiting time for the 1-customer. The servers have equal parameters $\Gamma_1^{\mathbb{X}} = \Gamma_2^{\mathbb{X}} = 1$ and $\alpha_1 = \alpha_2 = 2$. However, the first has a lower service rate than the second; in particular, $\mu_1 = 0.8 < 1 = \mu_2$. Clearly, in the worst case, the 2-customer has service priority.

Suppose first that all service times attain their worst-case values. In particular, servers 1 and 2 become available for service for the first time at $c_1^1 = x_1^1 = 1/\mu_1 + \Gamma_1^{\mathbb{X}}\sqrt{1} = 2.25$ and $c_2^1 = x_2^1 = 1/\mu_2 + \Gamma_2^{\mathbb{X}}\sqrt{1} = 2$, respectively. Then, at $t = 2$, server 2 starts servicing the 2-customer, and at $t = 2.25$, the 1-customer receives service. In other words, under worst-case service times, $W_1 = 2.25$.

Suppose now that the service times of server 1 are lower than their worst-case values. Specifically, server 1 takes $c_1^1 = x_1^1 = 1.8$ to become available for the first time. Then, at $t = 1.8$, server 1 starts servicing the 2-customer. At $t = 2$, server 2 will become available for service but will remain idle, being ineligible to serve the 1-customer. If server 1's time to serve the 2-customer takes its worst-case value such that $c_1^2 = x_1^1 + x_1^2 = 2/\mu_1 + \Gamma_1^{\mathbb{X}}\sqrt{2} = 2.5 + \sqrt{2}$, the 1-customer will be served precisely at that time and $W_1 = 2.5 + \sqrt{2} > 2.25$.

The intuition behind the counterexample is that while, on one the hand, shorter service times make the servers available earlier and could thus reduce wait times, on the other hand, they could also change the service sequence of customers, thus potentially increasing wait times for some customer classes. Our next result shows that the structure of hierarchical service systems precludes this latter possibility for customers waiting for service of any level.

Lemma 1. *For a hierarchical MCMS system, the clearing time $\mathcal{W}_K$ is increasing in the service times. In particular, problem (1) admits an optimal solution for which completion times take their worst-case values—i.e.,*

$$c_j^\ell = x_j^1 + \cdots + x_j^\ell = \frac{\ell}{\mu_j} + \Gamma_j^{\mathbb{X}}(\ell)^{1/\alpha_j}, \qquad j = 1, \ldots, K,\ \ell = 1, \ldots, \bar{\ell}_j.$$

Based on Lemma 1, we now fix the completion times to take their worst-case values. We introduce the following notation. Consider the set of all worst-case completion times for all servers—i.e., $\{c_j^\ell = \ell/\mu_j + \Gamma_j^{\mathbb{X}}(\ell)^{1/\alpha_j} : j = 1, \ldots, K,\ \ell = 1, \ldots, \bar{\ell}_j\}$[7]—and let $\bar{\ell} := \bar{\ell}_1 + \cdots + \bar{\ell}_K$ be its cardinality and c^ℓ its ℓth smallest element, $\ell = 1, \ldots, \bar{\ell}$. Consider the MIP

$$\text{maximize} \quad \sum_{\ell=2,\ldots,\bar{\ell}} c^\ell (f^{\ell-1} - f^\ell) \tag{3a}$$

subject to

$$\sum_{k=j,\ldots,K} y_{kj}^\ell \le 1, \quad \ell = 1, \ldots, \bar{\ell}_j,\ j = 1, \ldots, K; \tag{3b}$$

$$\sum_{(j,\omega):\, c_j^\omega \le c^\ell} y_{Kj}^\omega \le n_K - f^\ell, \quad \ell = 1, \ldots, \bar{\ell}; \tag{3c}$$

$$\sum_{\substack{j=1,\ldots,k \\ \ell=1,\ldots,\bar{\ell}_j}} y_{kj}^\ell \le n_k, \quad k = 1, \ldots, K-1; \tag{3d}$$

$$\sum_{\substack{(j,\omega):\, c_j^\omega = c^\ell \\ k=j,\ldots,K}} y_{kj}^\omega \ge f^\ell, \quad \ell = 1, \ldots, \bar{\ell}; \tag{3e}$$

$$f^{\ell-1} \ge f^\ell, \quad \ell = 2, \ldots, \bar{\ell}; \tag{3f}$$

$$f^\ell \in \{0,1\}, \quad \ell = 1, \ldots, \bar{\ell}; \tag{3g}$$

$$y_{kj}^\ell \in \{0,1\}, \quad j = 1, \ldots, K,\ k = j, \ldots, K,\ \ell = 1, \ldots, \bar{\ell}_j; \tag{3h}$$

$$n \in \mathbb{P} \cap \mathbb{N}^K, \tag{3i}$$

where $y \in \{0,1\}^{K\bar{\ell}_1 + (K-1)\bar{\ell}_2 + \cdots + \bar{\ell}_K}$ and $n \in \mathbb{N}^K$ are assignment and class population variables, respectively; and $f \in \{0,1\}^{\bar{\ell}}$ are indicators of whether the Kth customer class is filled, or has cleared.

Theorem 2. *For a hierarchical MCMS system, the optimal value of the MIP (3) is equal to W_K.*

Formulation (3) presents a structure that closely mimics an assignment problem. Constraints (3b) are classical assignment constraints. Constraints (3c) and (3d) are capacity constraints. The main departure from an assignment problem stems from the variable f^ℓ, which indicates whether the Kth customer class is filled, or has cleared, at the ℓth completion time, or equivalently, assignment. When that class clears, f^ℓ takes the value 0 (and retains it because of (3f)), allowing (3c) to be binding with n_K assignments to the Kth class, and the objective (3a) to attain the associated clearing time. Finally, (3e) forces assignment unless the Kth class has cleared, similarly to (2d).

Owing to its simpler structure and significantly fewer variables/constraints, we expect formulation (3) to yield significant computational advantages over the general formulation (2). We compare the two approaches in terms of their computational performance in Section 4.3, alongside a third heuristic approach, which we present next.

4.2. Service Wait Time Approximation

Both the formulation (2) for general MCMS and the more efficient formulation (3) for HMCMS systems have a number of variables that depends on the customer classes' populations. Intuitively, this is because the presence of more customers would require a higher number of server-to-customer assignments. Algebraically, as the population uncertainty set $\mathbb{P}$ includes higher-valued vectors n, the parameters $\bar{\ell}_j$ increase and so do the numbers of variables y and f. This dependence would increase computational burden for heavily overloaded systems. To overcome this, we devise a heuristic to approximate W_K with significantly reduced computational requirements that are independent of n. More importantly, we back the heuristic with an approximation guarantee that becomes tighter as n grows—i.e., precisely when the heuristic's computational gains become worthwhile.

Consider the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & w \\ \text{subject to} \quad & w \le \frac{m_j}{\mu_j} + \Gamma_j^{\mathbb{X}} s_j, \quad j = 1, \dots, K \\ & (s_j)^{\alpha_j} \le m_j, \quad j = 1, \dots, K; \\ & \sum_{k=j}^{K} m_k \le \sum_{k=j}^{K} n_k + K - j, \quad j = 1, \dots, K; \\ & n \in \mathbb{P}. \end{aligned} \tag{4}$$

It can be readily seen that problem (4) is convex. In particular, for any rational value of α_j (including the important case where the service times do not exhibit heavy tails; i.e., for $\alpha_j = 2$), problem (4) reduces to a second-order cone program (SOCP) (Alizadeh and Goldfarb 2001, section 2.3). An interpretation of its variables and constraints is as follows. The variables $m \in \mathbb{R}^K$ represent the numbers of customers assigned to/served by each server by the time the Kth class has cleared, which in turn corresponds to variable $w \in \mathbb{R}$. Variables $s \in \mathbb{R}^K$ are auxiliary and $n \in \mathbb{R}^K$ are class populations as before. At optimality, it can be readily seen that the first two constraints are equivalent with $w \le m_j/\mu_j + \Gamma_j^{\mathbb{X}}(m_j)^{1/\alpha_j}$—i.e., w is upper-bounded by the worst-case time it takes the jth server to serve its m_j assigned customers, for all $j = 1, \dots, K$. The third constraint bounds the number of customers assigned to a subset of servers by the population of customer classes these servers are eligible for, plus a correction term. Note that all variables are continuous and, as such, approximations of the quantities we just discussed.

In contrast with both our previous formulations, (4) can be interpreted as taking an "aggregate view" of the system, in that it only deals with the total number of customers served by each server, and not with which precise customer classes and in what order this occurred. Consequently, formulation (4) affords a drastic complexity reduction, falling into the category of conic optimization problems—namely, SOCP—that are efficient to solve at very high scale using standard solvers. Additionally, (4) involves only $3K + 1$ variables and a number of constraints that does not increase with the class populations n—unlike our previous formulations. Hence, in applications where n could take high enough values that render (3) impractical to solve, (4) provides an alternative approach.

Another implication of the aggregate system view of (4) is that it only provides an approximation to the quantity W_K we want to calculate. Fortunately, we are able to provide the following guarantee to the approximation fidelity. Specifically, the optimal value of (4), denoted by $\hat{W}_K$, approximates W_K within an additive constant that depends only on the maximum service time among all servers

$$\chi := \max_{j=1,\dots,K} \left\{ \frac{1}{\mu_j} + \Gamma_j^{\mathbb{X}} \right\}.$$

In particular, for $x \in \mathbb{X}$, we have that $x_j^\ell \le \chi$ for all $j = 1, \dots, K$ and $\ell = 1, \dots, \bar{\ell}_j$.

Theorem 3. *For a hierarchical MCMS system,*

$$W_K \le \hat{W}_K \le W_K + 2\chi.$$

A very important property of our approximation guarantee is that it becomes tighter as the class populations n increase—i.e., exactly for the problem instances for which formulation (4) would be most useful. To see this, note that as n increases, ceteris paribus, W_K also naturally increases as servers have to serve more customers. However, χ remains constant.

We next confirm by way of numerical studies that our heuristic approach yields significant computational benefits at essentially no cost in accuracy, as our approximation guarantee suggests.

4.3. Performance

We conclude the treatment of HMCMS systems with an evaluation of the two formulations we presented by way of numerical studies. In particular, we quantify, first, the required computation times of MIP (3) and the heuristic SOCP (4) (for $\alpha_j = 2$), relative to the general MIP formulation (2), and, second, the relative approximation error of the heuristic, $(\hat{W}_K - W_K)/W_K \times 100\%$.

We used a similar approach as in Section 3.3, randomly generating multiple problem instances of HMCMS systems of varying classes and population sizes. For a detailed discussion, see Appendix C.

Tables 3 and 4 summarize our findings. Specifically, Table 3 reports the average computation times of the three formulations under consideration for different problem sizes (as measured by the average total population sizes). Our results suggest that MIP (3) reduces computation times by a factor of three to four, approximately, compared to the general MIP formulation (2). The heuristic SOCP formulation provides a *further* reduction by a factor higher than 10.

Table 4 reports the average relative approximation errors we recorded for varying problem sizes. Evidently, our heuristic is almost exact and becomes tighter as population sizes grow.

Together, our findings from Tables 3 and 4 suggest that for problem sizes involving less than 10,000 customers, the exact MIP formulations can be used to produce solutions in a matter of two minutes. For problems involving a higher number of customers, the SOCP formulation retains the low computation times, with an approximation error of less than 0.1%.

In summary, the special structure of hierarchical service systems allowed us to sharpen our formulations to compute the wait time for service W_K. Formulation (3), by providing a speed increase by a factor of three to four, enables us to solve realistic-size problems, for example in the context of the kidney allocation system, involving 10 classes and 1,000 customers in approximately two minutes. We also provided a powerful heuristic that further reduced computational burden by an order of magnitude, and this allows us to preserve low computation time requirements for much larger instances, obtaining provably near-optimal solutions at the same time.

Table 3. Approximate Average Computation Times of Our Different Formulations for HMCMS Systems with Varying Number of Customers

Avg. total number of customers	Computation times		
	MIP (2)	MIP (3)	SOCP (4)
100 customers	1 sec	0.8 sec	0.8 sec
1,000 customers	< 1 min	$<\frac{1}{2}$ min	1.2 sec
10,000 customers	6 min	2 min	5.4 sec
100,000 customers	40 min	10 min	< 1 min

Table 4. Average Relative Approximation Error of Our SOCP Heuristic (4) for HMCMS Systems with Varying Number of Customers

Avg. total number of customers	Avg. relative error (%)
50 customers	1.9
100 customers	0.85
200 customers	0.5
400 customers	0.25
1,200 customers	0.08

5. Patient Wait Times in the U.S. Kidney Allocation System

In this section, we investigate a real-world application of our MCMS analysis framework. In particular, we consider the estimation of patient wait times in the U.S. kidney allocation system (KAS). We envision our methodology to enable transplant centers to develop software tools that would offer wait time estimates to their patients. We first describe the KAS in some detail and demonstrate that it effectively operates as a hierarchical service system. Then, we illustrate how our analysis framework can be deployed to estimate wait times, and conclude by performing a numerical case study based on historical data.

5.1. The U.S. Kidney Allocation System: An FCFS Hierarchical Service System

Kidney allocation in the United States is coordinated by the United Network for Organ Sharing (UNOS). When a patient is in need of a kidney transplant, his medical information is added to UNOSNet, a computerized system administered by UNOS. When a deceased-donor kidney is procured, the donor's information is also entered into the system. Subsequently, UNOSNet generates a *match run*—i.e., a ranked list of patients based on a set of allocation rules. The organ is then offered to the patient ranked highest on the match run. If rejected, it is offered to the patient ranked second highest, and so on. We next describe in some detail the allocation rules prevailing in the United States in the period from January 1, 2007, to January 2, 2014, for which we were able to obtain match run data. Note that some changes to the rules came into effect on December 4, 2014—these are discussed in Section 6 together with an extension to our methodology that can cater for these changes.

In KAS, the United States is divided geographically into 11 regions, each of which consists of several Organ Procurement Organizations (OPOs). There are a total of 58 OPOs of varying size. Before generating a match run, UNOSNet first screens out all medically incompatible candidates primarily based on blood type—other less frequent reasons could be height, weight, or tissue type.[8] Subsequently, the rank-ordered list is generated as follows. First, kidneys are offered to any identical tissue match candidates,[9] although such matches are extremely rare. Then, they are offered in turn to candidates in the same OPO as the donor, to candidates in the same region, and finally to all remaining candidates nationally. Within each classification, candidates are ranked using a points-based system, relying on (a) candidate wait time, (b) sensitization,[10] and (c) tissue match strength.

On receiving an offer, a patient is given an hour to decide whether to accept or reject it. Patients are more likely to reject lower-quality organs—e.g., organs from elderly donors or with a high creatinine level—because they would yield lower posttransplant survivability. In particular, the accept/reject decision involves trading off the benefits of an immediate transplant of the offered organ with the risks and benefits of waiting for future offers, whenever they might occur. In practice, some patients may be obliged to reject an offer because of operational reasons (e.g., patient is too ill for transplant, surgeon is unavailable, etc.); we shall refer to such patients as *unavailable*. Note that patients are able to observe only their rank in the match run, alongside donor information. Specifically, they have no information about any other candidate in the match run or the waitlist.

5.1.1. Modeling KAS as a Queuing System. The KAS can be reasonably approximated by a number of independent systems, each operating as a hierarchical MCMS queuing system under an FCFS priority. We elaborate on these modeling choices below. Note that these choices are hardly new and are in line with the literature, as we point out in the subsequent discussion and in Section 1.1.

In particular, we consider the patients and donors in a specific OPO and of a specific blood type as an independent system that we analyze separately. This is because patients predominantly accept kidneys from donors that are from the same OPO and of the same blood type. Indeed, kidneys are offered almost exclusively to candidates with identical blood type because of medical compatibility issues—exceptions arise in the extremely rare cases of identical tissue matches. Furthermore, the vast majority of candidates accept kidneys from their own OPO (close to 85%), finding kidneys from distant locations undesirable, owing to the procured organs' limited preservation times and their quality deterioration over (transport) time. Nonetheless, we illustrate how our model can be extended to capture coupling between different OPOs in Section 6.

The accept/reject decision-making process of candidates allows us to model each subsystem of an OPO–blood-type pair as a hierarchical MCMS queuing system. Specifically, there is a series of papers in the literature that model the accept/reject decision problem facing transplant patients as a stopping problem, where benefits from a current offer are traded off with risks of waiting and benefits from future potential offers. In that context, it has been shown that patients make decisions by following a *threshold*-based policy—i.e., they accept an offered kidney if and only if its quality exceeds a certain threshold, which depends on the patients' risk tolerance, health status, etc. (see Section 1.1). We assume that patients follow a threshold policy in our setting. Consequently, by clustering kidneys into levels $1, \ldots, K$ of decreasing quality, we can model the underlying dynamics with an HMCMS as follows: all waiting patients willing to accept service (kidneys) of quality level i or higher are assigned to class $i \in \{1, \ldots, K\}$. Correspondingly, there are K servers that capture the arrival processes of donated kidneys, with the jth server "producing" kidneys of quality level j and thus being eligible to serve patients of class $i \geq j$. When the jth server starts servicing a patient, this corresponds to a kidney of quality j being procured and accepted by the served patient, who then leaves the system. The server's service time corresponds to the time until the next kidney of quality j is procured.

Finally, it is well accepted both by practitioners and academics that candidates are ranked mostly in the order in which they joined the waitlist—i.e., the HMCMS queuing system in each OPO-blood-type subsystem essentially operates under an FCFS priority (see, e.g., Cleveland Clinic 2015, OPTNKTC 2007, Su and Zenios 2005). We note that while the KAS was originally designed in the 1980s so as to balance fairness (FCFS) and efficiency (stronger tissue matching), medical advances since the 1990s have drastically improved survivability under dialysis, to the extent that candidates have accumulated a large number of points from wait time that far outweigh other factors in the points computation and ultimately in their ranking (OPTNKTC 2007).

To summarize, the KAS can be credibly modeled as a collection of OPO-blood-type subsystems, each operating as an HMCMS system under FCFS—with patients corresponding to customers seeking service (transplantation), servers capturing the donation process, and service times corresponding to kidney interarrival times.

We next argue that our robust MCMS analysis framework is an appropriate solution method to adopt, because it accommodates practical considerations such as lack of information and instability. We will also see that our framework is flexible to account for other KAS dynamics we have not explicitly modeled, such as patient unavailability or removal from the waitlist because of death.

5.2. Using Our MCMS Analysis Framework

We propose using the robust MCMS analysis framework we have developed to estimate patient wait times in the KAS. By computing the clearing times for each queue, our model can essentially provide patients with estimates for the required wait time until they are offered an organ of the highest quality (W_1), or an organ of quality i or better (W_i), or simply any organ (W_K). More importantly, our model is suitable to provide credible estimates for all of the reasons we outlined in Section 3: the KAS is inherently unstable and plagued by incomplete information. Having discussed the former in Section 3, we elaborate on the latter below.

The key pieces of information that are unobservable in the KAS are the patients' preferences that drive accept/reject decisions, and these could significantly impact wait times. In particular, a specific patient observes only his rank in the match run, which informs him about how many patients are in front of him in the system. However, he is unable to know what organ qualities they would be willing to accept. If all patients in front of him were willing to accept only top-quality kidneys, he would likely get an offer sooner (of a lower-quality kidney); if they were willing to accept any kidney quality, he would likely wait much longer. To make things worse, fitting probabilistic prediction models of patient acceptance/rejection behavior has proved to be extremely challenging.[11]

In our terminology, while a patient could infer the aggregate queue population through his rank, there is significant uncertainty of how the population is distributed across the different queues. Our method is tailored to deal with this problem by taking a robust approach and by requiring the calibration of an uncertainty set, which is significantly easier compared to a probabilistic model.

It is important to note that modeling the queue populations via an uncertainty set allows us to capture other dynamics of the KAS that we do not explicitly model. For example, patients might become unavailable or might leave the system—e.g., because of death or receipt of an organ from a living donor. Also, patient preferences might change over time—e.g., again because of changes in their health condition. Patient rank might also be slightly affected by tissue matching and sensitization, resulting in fewer patients with higher priority. All of these aforementioned dynamics would affect the queue populations and could thus be subsumed by properly calibrated uncertainty sets.

5.2.1. Model Calibration.

We cluster kidneys based on the well-accepted Kidney Donor Profile Index (KDPI), a quality metric that UNOS has adopted.[12] Although in practice physicians and patients might be assessing quality in ways that deviate from KDPI slightly, Arıkan et al. (2018) brought forth empirical evidence that accept/reject quality thresholds can be well approximated by KDPI.

With regard to the queue population uncertainty set, we specify the set in a way that it relies only on parameters that can be estimated through available data, so as to retain practical relevance. Consider a k-patient who observes his rank to be r—i.e., there are $r-1$ patients in front of him. Let $\mathcal{Z}_\nu$ be the class to which the νth such patient belongs, $\nu = 1, \dots, r-1$. In case the νth patient is unavailable, we let $\mathcal{Z}_\nu = 0$. Let q_i be the probability of a patient being of class $i \in \{1, \dots, K\}$, or being unavailable ($i = 0$). That is, $\mathcal{Z}_\nu = i$ with probability q_i, for all $i = 0, 1, \dots, K$ and $\nu = 1, \dots, r-1$. Assuming independence, a CLT-based approximation would then yield that

$$\sum_{\nu=1}^{r-1} \mathcal{Z}_\nu - (r-1)\mu_{\mathcal{Z}} \le \Gamma \sigma_{\mathcal{Z}} \sqrt{r-1},$$

where $\mu_{\mathcal{Z}} = \sum_{i=1}^{K} i q_i$, $\sigma_{\mathcal{Z}}^2 = \sum_{i=1}^{K} i^2 q_i - \mu_{\mathcal{Z}}^2$, and Γ is a conservatism parameter. Noticing that $\sum_{\nu=1}^{r-1} \mathcal{Z}_\nu + k = \sum_{i=1}^{K} i n_i$, we get that

$$\mathbb{P} = \left\{ n \in \mathbb{R}^K \colon \sum_{i=1}^{K} i n_i - k \le (r-1)\mu_{\mathcal{Z}} + \Gamma \sigma_{\mathcal{Z}} \sqrt{r-1} \right\}. \quad (5)$$

5.3. Numerical Case Study

In this study, we apply our robust MCMS (RMCMS) methodology to estimate wait times statistics in dependence of rank for patients of blood type O in the PADV-OP1 Gift of Life Donor Program[13] OPO. So as to test our methodology in a realistic setting, we obtained all historical data from UNOS that would be available to patients and their physicians. We split the data into a *training set*, used to fit model parameters, and a *testing set*, used to assess out-of-sample performance.

5.3.1. Data.

Our data set covers the period from May of 2007 to June of 2013 and includes 7,388 patients and 438 donors. We use the data from May of 2007 to May of 2010 as our training set, and the remainder as our testing set. The data set includes the following information pertaining to each procured deceased-donor kidney: (a) procurement OPO, (b) procurement date and time, (c) donor blood type, (d) KDPI score, and (e) all accept/reject decisions made, alongside reasons for rejection, e.g., due to quality or unavailability.

It is important to note here that our data set also includes, for each offer made, patient identifiers. These identifiers enable us to reconstruct the entire sequence of offers received by each patient, and thus to compute their individual wait times. However, because of confidentiality reasons, this information is made available by UNOS only for bona fide research purposes under institutional review board oversight. In particular, it would not be available to patients or physicians for consultation purposes. As such, we do not use this identifier information in any way in our parameter fitting process. Instead, we only use it for purposes of evaluating our implementation's accuracy. In other words, our implementation here relies on publicly available data only and can be replicated by transplant centers wishing to offer consultation to their patients.

This study used data from the Organ Procurement and Transplantation Network (OPTN). The OPTN data system includes data on all donor, wait-listed candidates, and transplant recipients in the United States, submitted by the members of the Organ Procurement and Transplantation Network (OPTN), and has been described elsewhere. The Health Resources and Services Administration (HRSA) of the U.S. Department of Health and Human Services provides oversight to the activities of the OPTN contractor.

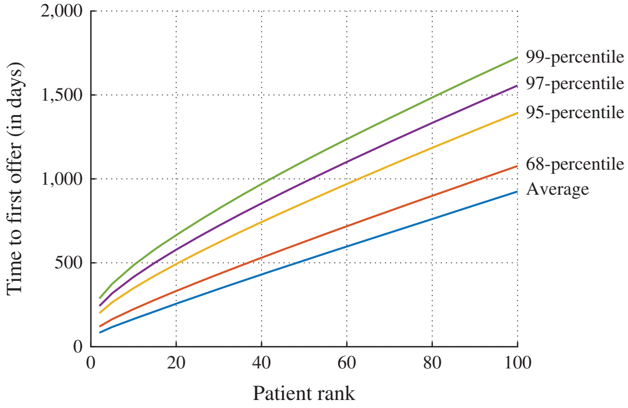
5.3.2. Parameter Fitting. We clustered kidneys in $K = 5$ quality categories based on KDPI, which is a normalized score from 0% (best quality) to 100% (worst quality). The categories $j = 1, \ldots, 5$ included all kidneys with a KDPI score of 0%–6%, 6%–25%, 25%–50%, 50%–75%, and 75%–100%, respectively.[14]

We used the kidney interarrival times in the training set to fit the service time uncertainty set parameters. In particular, we set the coefficient $\alpha_j = 2$ for all quality categories, based on the absence of heavy tails in the empirical distributions. For the j-quality kidneys, we let $1/\mu_j$ equal the interarrivals' empirical mean. Similarly, we let $\Gamma_j^{\mathbb{X}} = \Gamma\sigma_j$, where σ_j equals the interarrivals' empirical standard deviation, and Γ is the same conservatism parameter as in (5).

For the queue population uncertainty set, we let q_0 be the empirical mean of the fraction of rejections due to unavailability in the training set. To estimate the probability q_i of a patient being of class $i \in \{1, \ldots, K\}$, we used a maximum likelihood approach. That is, we fitted the probabilities q_i's so as to maximize the likelihood of the accept/reject decisions we observed in the training set—we refer the reader to Appendix D for more details.

5.3.3. Out-of-Sample Performance. Having fitted all parameters based on the training set, we used our SOCP (4) with various values of the conservatism parameter Γ (as discussed in Section 3.3) to estimate the average, 68-, 95-, 97-, and 99-percentiles of the wait time for blood group O patients in the PADV-OP1 Gift of Life Donor Program OPO in the testing set, depending on their rank. Our estimates are depicted in Figure 3. For example, we estimate the average wait time that a patient ranked 50 will experience until he is offered an organ to be approximately 500 days.

Figure 3. (Color online) Our Model's Estimates of Different Statistics of Time to First Offer vs. Patient Rank in a Particular OPO and Blood Group



To evaluate the accuracy of our estimates out-of-sample, we used the patient identifier information in our data set to empirically calculate statistics for the wait times actually experienced by patients in our testing set. Because of limited data availability, we were only able to credibly calculate the average and 68-percentile for patients ranked up to 40. Table 5 includes the empirical estimates, together with our RMCMS model's estimates. The average absolute errors of our estimates relative to the empirical ones were 14.96% for the average and 11.73% for the 68-percentile.

For benchmark purposes, we consider a *hypothetical* estimator that uses additional historical patient wait time information, and thus refer to it as "historical." In particular, we estimate the average (68-percentile) wait time of a patient of a given rank in the testing set by the average (68-percentile) historical wait time of patients of the same rank in the training set. This estimator is inspired by the so-called "delay history estimators" studied in queuing theory (Ibrahim et al. 2017). We referred to such an estimator as hypothetical in this context because historical wait time information is *not* available to patients or physicians as per our discussion above. In other words, the historical estimator could not be deployed in practice. Another significant limitation of the hypothetical historical estimator compared to our model is that it can only provide estimates up to some rank and up to some percentiles for which enough historical data are available. Consequently, we were only able to use it to estimate average (68-percentile) wait times for patients ranked up to 40 as before. The average absolute errors of the historical estimator relative to the empirical estimates were 16.76% for the average and 14.65% for the

Table 5. Statistics of Time to First Offer in Dependence of Patient Rank in a Particular OPO and Blood Group

	Average (in days)			68-percentile (in days)		
Rank	Empirical	Historical[a]	RMCMS	Empirical	Historical[a]	RMCMS
1–5	110.00	71.50	100.45	178.24	122.80	141.90
5–10	133.00	128.00	141.54	209.84	235.40	193.70
10–15	243.00	188.50	188.63	328.86	349.00	251.27
15–20	308.50	235.00	234.07	405.74	383.76	305.37
20–25	292.00	335.50	278.50	345.38	436.72	357.38
25–30	319.00	300.00	322.23	409.60	450.80	407.94
30–35	261.00	272.00	365.41	444.82	468.06	457.40
35–40	363.00	450.00	408.17	457.00	551.30	506.00
Avg. abs. rel. error across all ranks (%)	0.00	16.76	14.96	0.00	14.65	11.73

Notes. Empirical wait times correspond to the actual wait times exhibited in the testing set. RMCMS (resp. historical) estimates correspond to the estimates obtained by our (resp. the historical estimator) approach.

[a] Historical estimator relies on data that are not publicly available and is provided for reference purposes only (see Section 5.3).

68-percentile. In contrast, our approach requires only publicly available data, is implementable in practice, generalizes to arbitrarily high ranks, and, despite using significantly less data, provides higher accuracy.

6. Class-Based Priority Systems

So far, we focused on MCMS systems that serve customers according to FCFS. We now extend our analysis to cater for two alternative priority rules that are frequently encountered in practice. In Section 6.1, we study systems in which customer priority is driven by the class to which they belong. In Appendix E, we study systems in which some servers prioritize customers based on their class, while others do so based on FCFS. In both cases, same-class customers are served according to FCFS.

The priority rules we consider here are motivated by practice. In particular, they arise in the U.S. kidney allocation system owing to a recent allocation policy change that came in effect in December 2014.[15] According to it, the new KAS offers the top-20% quality kidneys (as measured by their KDPI; see Section 5) to patients with top-20% expected posttransplant survival (EPTS) score first, and then to the remaining patients.[16] That is, patient priority for top-quality kidneys is driven by whether they belong to the top-20% EPTS class or not, whereas the remaining kidneys are offered in an FCFS manner. In Appendix E, we show how the formulation we developed in Section 5.2 to estimate wait times in the KAS can be extended to capture this policy change.

Similarly, class-based priority rules can be used to model regional or national kidney offers. In particular, our KAS model in Section 5.2 ignored such offers and treated each local OPO independently (since that accounted for the vast majority of transplants). To further enhance our estimates, one can envision a national KAS model with 58 MCMS systems of the type we studied in Section 5.2, each corresponding to one of the 58 OPOs. Procured kidneys would then be offered to patients within the same OPO first, then to patients within the same region, and then to the remaining patients—i.e., different patient classes would have different priorities.

Finally, various house allocation programs prioritize applicants based on additional criteria to wait time—e.g., the Housing Authority in Cambridge, Massachusetts, prioritizes those who either live/work in Cambridge or are veterans—while serving based on FCFS otherwise. Class-based priority rules become relevant under such circumstances.

6.1. Class Priority Systems

We study the alternative priority rules only for hierarchical service systems. This is due to space considerations, but it also allows us to keep our focus on the paper's main application, the KAS. General MCMS systems under class-based priority rules can be analyzed in a similar fashion. Our treatment parallels the one we presented for MCMS FCFS systems.

6.1.1. Model Dynamics. Consider an HMCMS system, where a customer's service priority is dictated by the class to which he belongs. In particular, there is a class priority ranking, so that customers from a higher-ranked class have priority over customers from lower-ranked classes. Customers from within a particular class are served in an FCFS manner. We henceforth refer to this service priority rule as *class priority* (CP). For simplicity, we present the case here where the priority rank of each class corresponds to its index—i.e.,

i-customers have service priority over k-customers, for all $i < k$.

In this context and for the purposes of computing wait times, neither the precise arrival order σ of customers waiting at $t = 0$ is needed, nor is the precise constellation of queues' populations $\mathbb{L}_i(t)$. Instead, it can be readily seen that a sufficient state representation is now given by the population size $|\mathbb{L}_i(t)|$ of each queue at time t, where $|\mathbb{L}_i(0)| = \mathcal{N}_i$, $i = 1, \ldots, K$. Then, if the jth server becomes available at time t, it serves a customer from class $i^\star \in \arg\min\{i \in \mathbb{Q}(j)\colon |\mathbb{L}_i(t)| > 0\}$ and, subsequently, $|\mathbb{L}_{i^\star}(t+)| = |\mathbb{L}_{i^\star}(t)| - 1$. If $|\mathbb{L}_i(t)| = 0$ for all $i \in \mathbb{Q}(j)$, then the server serves a customer of an external class, assumed to always be populated.[17]

Suppose we are interested in quantifying the wait time of an i-customer. As before, we assume that no i-customers arrive after $t = 0$, because future i-customers would not affect wait times of existing ones. This no longer being true for customers of higher priority classes $1, \ldots, i-1$, we explicitly model such arrivals. In particular, k-customers arrive at an average rate λ_k after $t = 0$, with i.i.d. interarrival times that are also independent of customer arrivals of other classes, service times, and queue populations, for all $k = 1, \ldots, i-1$. We denote the arrival time of the rth k-customer after $t = 0$ with $\mathcal{A}_k^r$, $k = 1, \ldots, i-1$, $r \in \mathbb{N}$ (in which case $|\mathbb{L}_k(\mathcal{A}_k^r+)| = |\mathbb{L}_k(\mathcal{A}_k^r)| + 1$).

All other dynamics and model parameters are as in Section 2. The clearing time of the ith customer class, defined as

$$\mathcal{W}_i^{\mathrm{CP}}(\mathcal{N}_1, \ldots, \mathcal{N}_K, \{\mathcal{X}_1^\ell\}_{\ell\in\mathbb{N}}, \ldots, \{\mathcal{X}_K^\ell\}_{\ell\in\mathbb{N}},\\ \{\mathcal{A}_1^r\}_{r\in\mathbb{N}}, \ldots, \{\mathcal{A}_{i-1}^r\}_{r\in\mathbb{N}}) := \inf\{t \geq 0\colon |\mathbb{L}_i(t)| = 0\},$$

can be used to analyze wait times for customers as per our discussion in Section 2. As a technical remark, note that for finite service times and queue populations, $\mathcal{W}_i^{\mathrm{CP}}$ will remain finite—in fact, since no i-customers arrive, class i will clear by the time the ith server serves $\mathcal{N}_i$ customers.

6.1.2. Model of Uncertainty.

To quantify $\mathcal{W}_i^{\mathrm{CP}}$, we assume that service times and queue populations lie in uncertainty sets $\mathbb{X}_j$ and $\mathbb{P}$ as in Section 3. Customer arrival times, being summations of i.i.d. interarrival times, are assumed to lie in GCTL-based uncertainty sets in accordance with the literature. In particular, k-customers' arrival times lie in the polyhedron

$$\mathbb{A}_k := \left\{a_k \in \mathbb{R}^{\bar{r}_k}\colon a_k^r \geq r/\lambda_k - \Gamma_k^{\mathbb{A}}(r)^{1/\beta_k},\ r = 1, \ldots, \bar{r}_k\right\},\\ k = 1, \ldots, i-1,$$

where $\Gamma_k^{\mathbb{A}}$ is a conservatism parameter,[18] β_k a heavy tail parameter, and $\bar{r}_k$ is the maximum number of arrivals (in a similar fashion as $\bar{\ell}_j$). As we shall see, a characterization of $\bar{r}_k$ would be superfluous.

6.1.3. Solution Methodology.

We quantify the clearing time $\mathcal{W}_i^{\mathrm{CP}}$ with a worst-case guarantee on its value, denoted by W_i^{CP} and given as the optimal value of the problem

$$\begin{aligned} \text{maximize}\quad & \mathcal{W}_i^{\mathrm{CP}}(n_1, \ldots, n_K, x_1, \ldots, x_K, a_1, \ldots, a_{i-1})\\ \text{subject to}\quad & n \in \mathbb{P} \cap \mathbb{N}^K;\\ & x_j \in \mathbb{X}_j, \quad j = 1, \ldots, K;\\ & a_k \in \mathbb{A}_k, \quad k = 1, \ldots, i-1. \end{aligned} \tag{6}$$

One can readily adapt the proof of Proposition 1 to show that (6) remains $\mathcal{NP}$-hard. Similarly to our analysis in Section 4, our first nontrivial result on hierarchical service systems under CP shows that in the problem above, we can take service and arrival times be equal to their worst-case values—i.e., have servers take as long as possible to serve and customers arrive as early as possible.

Lemma 2. *For a hierarchical service system under CP, the clearing time $\mathcal{W}_i^{\mathrm{CP}}$ is increasing in the service times and decreasing in the arrival times. In particular, problem (6) admits an optimal solution for which completion and arrival times take their worst-case values—i.e.,*

$$c_j^\ell = x_j^1 + \cdots + x_j^\ell = \frac{\ell}{\mu_j} + \Gamma_j^{\mathbb{X}}(\ell)^{1/\alpha_j}, \quad j = 1, \ldots, K, \ell = 1, \ldots, \bar{\ell}_j;$$

$$a_k^r = \frac{r}{\lambda_k} - \Gamma_k^{\mathbb{A}}(r)^{1/\beta_k}, \quad k = 1, \ldots, i-1, r = 1, \ldots, \bar{r}_k.$$

Taking advantage of Lemma 2, we fix the completion and arrival times to their worst-case values. We next formulate an MIP to compute W_i^{CP} that is similar to the efficient formulation (3), resembling an assignment problem. Recall that c^ℓ is the ℓth smallest element of the set comprising of all completion times c_j^l, for all $l = 1, \ldots, \bar{\ell}_j$ and $j = 1, \ldots, K$. Let v_k^ℓ be the number of k-customer arrivals by time c^ℓ—i.e., $v_k^\ell := \max\{r\colon a_k^r \leq c^\ell\}$ for $k = 1, \ldots, i-1$ and $\ell = 1, \ldots, \bar{\ell}$.

Consider the problem

$$\text{maximize} \quad \sum_{\ell=2,\ldots,\bar{\ell}} c^\ell(f_i^{\ell-1} - f_i^\ell) \tag{7a}$$

subject to

$$\sum_{k=j,\ldots,K} y_{kj}^\ell \leq 1, \quad \ell = 1, \ldots, \bar{\ell}_j,\ j = 1, \ldots, K; \tag{7b}$$

$$\sum_{(j,\omega)\colon c_j^\omega \leq c^\ell} y_{ij}^\omega \leq n_i - f_i^\ell, \quad \ell = 1, \ldots, \bar{\ell}; \tag{7c}$$

$$\sum_{(j,\omega)\colon c_j^\omega \leq c^\ell} y_{kj}^\omega \leq n_k + v_k^\ell - f_k^\ell,\\ k = 1, \ldots, i-1,\ \ell = 1, \ldots, \bar{\ell}; \tag{7d}$$

$$\sum_{\substack{(j,\omega)\colon c_j^\omega = c^\ell\\ k=j,\ldots,i}} y_{kj}^\omega \geq f_i^\ell, \quad \ell = 1, \ldots, \bar{\ell}; \tag{7e}$$

$$f_i^{\ell-1} \geq f_i^\ell, \quad \ell = 2, \ldots, \bar{\ell}; \tag{7f}$$

$$y_{kj}^\omega \leq 1 - f_{k'}^\ell, \quad k' < k,\ (j,\omega)\colon c_j^\omega = c^\ell,\ \ell = 1, \ldots, \bar{\ell}; \tag{7g}$$

$$f_k^\ell \in \{0,1\}, \quad k=1,\dots,i,\ \ell=1,\dots,\bar{\ell}; \tag{7h}$$
$$y_{kj}^\ell \in \{0,1\}, \quad j=1,\dots,K,\ k=j,\dots,K,\ \ell=1,\dots,\bar{\ell}_j; \tag{7i}$$
$$n \in \mathbb{P} \cap \mathbb{N}^K. \tag{7j}$$

Theorem 4. *For a hierarchical MCMS system under class priority, the optimal value of the MIP (7) is equal to W_i^{CP}, $i=1,\dots,K$.*

The MIP (7) is very similar to (3) (for $i=K$), with its variables and constraints having the same interpretation. The only two discrepancies are as follows. First, in this setting, customer arrivals are possible. This is reflected in (7d), where the number of assigned services to the kth class is bounded by its initial population n_k adjusted for arrivals v_k^ℓ. Second, in this case, the priority discipline dictates that k'-customers have priority over k-customers, for all $k' < k$. To capture this, we use variables f_k^ℓ that indicate whether class k is filled or has cleared by time c^ℓ. Constraint (7g) enforces then the CP discipline: if at c^ℓ the k'th class is filled, the server cannot be assigned to any lower-priority $k > k'$ class—i.e., $y_{kj}^\omega \le 1 - f_{k'}^\ell = 0$.

As a technical remark, the parameters $\bar{\ell}_j$ can be calculated as follows. First, note that the ith class must have cleared after the ith server has served n_i customers, since i-customers have priority among the ones for which the ith server is eligible. Thus, $\bar{\ell}_i = \max\{n_i : n \in \mathbb{P} \cap \mathbb{N}^K\}$. At the same time, this observation implies that the ith class must have cleared by $c_i^{\bar{\ell}_i}$, which is precisely the time it takes the ith server to serve its maximum amount of customers. The parameters $\bar{\ell}_j$ can then be taken as the maximum customers each server could serve within $c_i^{\bar{\ell}_i}$—i.e., $\bar{\ell}_j = \max\{l : c_j^l \le c_i^{\bar{\ell}_i}\}$.

6.1.4. Scalable Approximation.

We conclude our analysis of CP hierarchical service systems by devising a heuristic that approximates W_i^{CP}. The heuristic is inspired by the "aggregate" allocation view that we discussed in Section 4.2. Specifically, consider the following convex optimization problem:

$$\begin{aligned}
&\text{maximize } w\\
&\text{subject to}\\
&w \le m_j/\mu_j + \Gamma_j^{\mathbb{X}} s_j, \quad j=1,\dots,i;\\
&(s_j)^{\alpha_j} \le m_j, \quad j=1,\dots,i;\\
&\sum_{k=j}^{i} m_k \le \sum_{k=j}^{i} n_k + \sum_{k=j}^{i-1} q_k + i - j, \quad j=1,\dots,i;\\
&q_j/\lambda_j - \Gamma_j^{\mathbb{A}} u_j \le w, \quad j=1,\dots,i-1;\\
&(u_j)^{\beta_j} \le q_j, \quad j=1,\dots,i-1;\\
&n \in \mathbb{P}.
\end{aligned} \tag{8}$$

The problem shares many similarities with (4), reducing to an efficient SOCP formulation for CLT-based uncertainty sets (when α_j and β_j are rational for all j). Here, variables $q \in \mathbb{R}^{i-1}$ capture customer arrivals. Accordingly, the number of customers assigned to a subset of servers is now bounded by the population of customer classes for which these servers are eligible, adjusted for arrivals. Variables $u \in \mathbb{R}^{i-1}$ are auxiliary and ensure that the customer arrivals q attain their appropriate value—i.e., the worst-case number of arrivals by the clearing time w.

Note that, in comparison with the MIP (7), this heuristic has significantly reduced computational requirements that are also independent of n. Next, we provide an approximation guarantee, for the special case when there are no arrivals, that becomes tighter as n grows in the same way as our heuristic in Section 4.2. Let $\hat{W}_i^{\text{CP}}$ be the optimal value of (8).

Theorem 5. *For a hierarchical MCMS system under class priority and no arrivals,*

$$W_i^{\text{CP}} \le \hat{W}_i^{\text{CP}} \le W_i^{\text{CP}} + 2\chi, \quad i=1,\dots,K.$$

Theorem 5 shows that our heuristic produces near-optimal results, for high value of n and the special case of no arrivals. This suggests that the heuristic will still provide quality approximations in the general case. In numerical studies we conducted, similar to the ones we presented in Section 4.3, we found the approximation errors, even under customer arrivals, to be no worse that the ones we reported in Section 4.3. We omit further details because of space limitations.

7. Conclusions

We dealt with the problem of estimating wait times in multiclass, multiserver (MCMS) queuing systems that operate based on predetermined priority rules under incomplete information. In particular, we focused on MCMS systems under FCFS, motivated by the U.S. kidney allocation system (KAS). To deal with primitive information incompleteness and the transient/unstable behavior that characterizes such systems, we developed a novel robust optimization framework. The framework was based on the introduction of an assignment-style formulation to capture the complex queuing dynamics in an MCMS system.

We devised MIP formulations for our estimation problem. We also presented a provably near-optimal heuristic that involved the solution of an SOCP for problems attaining a particular hierarchical structure, commonly encountered in practice.

To validate the performance of our approach in terms of computation times and accuracy, we performed numerical studies in which we found our method to significantly outperform simulation. We also presented an implementation in the context of the KAS. Specifically, we calibrated our model so as to estimate wait times of patients based on their own unique characteristics, preferences, and information available. Using detailed historical data, we fitted our model parameters and measured the out-of-sample estimation error to be less compared to hypothetical estimators that utilized data not available to patients. To the best of our knowledge, such an estimation tool is novel and can provide valuable information to patients as they plan their treatment options and life activities. Furthermore, we analyzed systems that operated under an alternatively priority rule, based on class priority, to illustrate how our framework can be generalized.

Acknowledgments

The authors thank the review team for the valuable feedback they provided. The data reported here have been supplied by United Network for Organ Sharing (UNOS) as the contractor for the Organ Procurement and Transplantation Network (OPTN). The interpretation and reporting of these data are the responsibility of the authors and in no way should be seen as an official policy of or interpretation by the OPTN or the U.S. Government.

Appendix A. Service Time Uncertainty Sets

The service time uncertainty sets in this paper are given by

$$\mathbb{X}_j := \left\{ x_j \in \mathbb{R}^{\bar{\ell}_j}\colon \sum_{k=1}^{\ell} x_j^k \le \frac{\ell}{\mu_j} + \Gamma_j^{\mathbb{X}} (\ell)^{1/\alpha_j},\ \ell = 1, \dots, \bar{\ell}_j \right\}, \quad j = 1, \dots, M,$$

where $\Gamma_j^{\mathbb{X}} \ge 0$ controls the degree of conservatism and $\alpha_j \in (1, 2]$ is a heavy tail parameter. We remark on how our choice of service time uncertainty sets and their structure affect our results, and possible ways to calibrate the sets using data and probabilistic guarantees. For an elaborate motivation and justification based on limit theorems, we refer the interested reader to Bandi and Bertsimas (2012) and Bandi et al. (2015).

A.1. Theoretical Results

It can be readily seen that all of our theoretical results in Section 3 extend in case the service time uncertainty sets $\mathbb{X}_j$ are nonempty, bounded polyhedra, for every $j = 1, \dots, M$. In particular, the proofs of our hardness result (Proposition 1) and MILP reformulation of problem (1) (Theorem 1) do not rely on the GCLT structure imposed by Assumption 1. Similarly, our monotonicity result in Lemma 1 holds more generally. The sharper formulations we derive in Section 4 for hierarchical service systems, however, do rely on properties of the GCLT structure (Theorems 2 and 3).

A.2. Constraints Structure

A more general way to formulate constraints based on GCLT is to consider a subset of service times, $S \subset \{1, \dots, \bar{\ell}_j\}$, and bound their sum as

$$\sum_{k \in S} x_j^k \le \frac{|S|}{\mu_j} + \Gamma_j^{\mathbb{X}} |S|^{1/\alpha_j}.$$

In our work, we imposed constraints that correspond to *nested* subsets of the form $S = \{1, \dots, \ell\}$ only (Assumption 1). Variations of this nested structure have been used in numerous papers in the robust optimization literature across different application areas, including, for example, Bandi et al. (2015), Whitt and You (2018), and Whitt and You (2016) (queuing), and Mamani et al. (2017) (inventory management).

Nonetheless, we argue next that all of our main results that rely on the GCLT structure, namely Theorems 2 and 3, still hold true if we consider sets that are generated by all possible GCLT-based constraints, specifically

$$\tilde{\mathbb{X}}_j := \left\{ x_j \in \mathbb{R}^{\bar{\ell}_j}\colon \sum_{k \in S} x_j^k \le \frac{|S|}{\mu_j} + \Gamma_j^{\mathbb{X}} |S|^{1/\alpha_j},\ \forall\, S \subset \{1, \dots, \bar{\ell}_j\} \right\}.$$

To show this, it suffices to show that the worst-case service times over the sets $\mathbb{X}_j$ we identify in Lemma 1, which we denote here by

$$\tilde{x}_j^{\ell} = \frac{1}{\mu_j} + \Gamma_j^{\mathbb{X}} (\ell^{1/\alpha_j} - (\ell - 1)^{1/\alpha_j}), \quad \ell = 1, \dots, \bar{\ell}_j,$$

remain feasible for $\tilde{\mathbb{X}}_j \subset \mathbb{X}_j$. To this end, consider all possible index sets of some fixed cardinality $\Delta \in \{1, \dots, \bar{\ell}_j\}$. We have

$$\sum_{k \in S} \tilde{x}_j^k \le \sum_{k=1}^{\Delta} \tilde{x}_j^k \le \frac{\Delta}{\mu_j} + \Gamma_j^{\mathbb{X}} \Delta^{1/\alpha_j}, \quad \forall\, S \subset \{1, \dots, \bar{\ell}_j\} \text{ such that } |S| = \Delta,$$

where the first inequality follows from $\tilde{x}_j^1 \ge \cdots \ge \tilde{x}_j^{\bar{\ell}_j}$, and the second from $\tilde{x}_j \in \mathbb{X}_j$. Thus, $\tilde{x}_j \in \tilde{\mathbb{X}}_j$.

A.3. Calibration Using Historical Data and Probabilistic Bounds

In this section, we discuss a possible way to calibrate the uncertainty set $\mathbb{X}_j$ for the important case wherein service times have finite variance and do not exhibit heavy tails. This is the case, for example, in the kidney allocation system, or when service times are exponentially distributed. That is, we set $\alpha_j = 2$. We can also set the mean service time $1/\mu_j$ equal to its empirical mean, calculated from available historical data—see, for example, the Parameter Fitting paragraph of Section 5.3. Similarly, we calculate the empirical standard deviation σ_j using data.

A possible way to calibrate the conservatism parameter $\Gamma_j^{\mathbb{X}}$ it to use probabilistic bounds as follows. We assume that service times follow some (unknown) distribution $\mathbf{P}$, and propose to use (approximate) probabilistic bounds to calibrate $\Gamma_j^{\mathbb{X}}$ so that service times lie in the uncertainty set with some prespecified confidence level. For technical purposes, we also require $\mathbf{P}$ to have a uniformly bounded third absolute moment.

The key idea is to notice that the constraints in $\mathbb{X}_j$ can be equivalently rewritten involving the maximum of a normalized random walk. In particular, if we let

$$M_{\bar{\ell}_j} := \max_{1 \le \ell \le \bar{\ell}_j} \frac{\sum_{k=1}^{\ell} ((x_j^k - 1/\mu_j)/\sigma_j)}{\sqrt{\ell}},$$

then we have that

$$\mathbb{X}_j = \left\{ x_j \in \mathbb{R}^{\bar{\ell}_j}\colon M_{\bar{\ell}_j} \le \Gamma_j^{\mathbb{X}} / \sigma_j \right\}.$$

If we now consider the associated random service times $\mathcal{X}_j^k$, it can be readily seen that $\mathcal{Y}_k := (\mathcal{X}_j^k - 1/\mu_j)/\sigma_j$ are independent, zero-mean, and unit-variance random variables. Letting $\mathcal{S}_\ell := \sum_{k=1}^{\ell} \mathcal{Y}_k$, we can write the random variable associated with $M_{\bar{\ell}_j}$ as

$$\mathcal{M}_{\bar{\ell}_j} := \max_{1 \le \ell \le \bar{\ell}_j} \frac{\mathcal{S}_\ell}{\sqrt{\ell}}.$$

Using this notation, we get that the probability of service times $\mathcal{X}_j^k$ lying in $\mathbb{X}_j$ is precisely

$$\mathbf{P}(\{\mathcal{X}_j^1, \dots, \mathcal{X}_j^{\bar{\ell}_j}\} \in \mathbb{X}_j) = \mathbf{P}\left(\mathcal{M}_{\bar{\ell}_j} \le \Gamma_j^{\mathbb{X}}/\sigma_j\right).$$

Using Theorem 1 in Darling and Erdős (1956), we get that for large enough $\bar{\ell}_j$,

$$\mathbf{P}\left(\mathcal{M}_{\bar{\ell}_j} \le \delta_{\bar{\ell}_j} + \frac{t}{\theta_{\bar{\ell}_j}}\right) \approx \exp\left(\frac{-\exp(-t)}{2\sqrt{\pi}}\right), \quad \forall\, t \in \mathbb{R},$$

where $\theta_n := \sqrt{2 \log\log n}$ and $\delta_n := \theta_n + (\log\log\log n)/(2\theta_n)$, $n \ge 1$. Therefore,

$$\mathbf{P}\left(\mathcal{M}_{\bar{\ell}_j} \le \frac{\Gamma_j^{\mathbb{X}}}{\sigma_j}\right) \approx \exp\left(\frac{-\exp(\theta_{\bar{\ell}_j}(\delta_{\bar{\ell}_j} - \Gamma_j^{\mathbb{X}}/\sigma_j))}{2\sqrt{\pi}}\right).$$

Hence, we conclude that if we want the service times to lie in the uncertainty set $\mathbb{X}_j$ with probability $1-\epsilon$, approximately, we can select

$$\Gamma_j^{\mathbb{X}} = \sigma_j \delta_{\bar{\ell}_j} - \frac{\sigma_j}{\theta_{\bar{\ell}_j}} \log\left(2\sqrt{\pi} \log \frac{1}{1-\epsilon}\right).$$

Appendix B. Numerical Experiments on Synthetic Instances of MCMS Systems

We performed two sets of experiments on an array of randomly generated instances of MCMS systems. In the first (second) set of experiments, we operate in a regime where the true distributions of queue populations are known (unknown). We note that the second setting is most relevant for the class of problems that we focus on in this paper.

B.1. Known Queue Population Distribution

When the distributions of all uncertain parameters are perfectly known, the clearing time distribution can be estimated using simulation. We estimate clearing time percentiles using our method and benchmark against simulation (assumed to return the true statistics). The following procedure underlies all of our experiments in this regime:

- Select $K = M$. Select also the mean $\mu^{\mathbb{P}}$ of each queue's population distribution. The populations of all queues are independent and normally distributed with standard deviation $\sigma^{\mathbb{P}} = 0.2$. Finally, select the distributions of the service times. These have mean $1/\mu_j = 1$ for all $j = 1, \dots, K$, and are either normally distributed with standard deviation σ_j or Pareto distributed with parameter α. Holding these parameters fixed, generate 100 instances of the problem by constructing server eligibility sets $\mathbb{S}$ at random. For each instance, select a queue index i uniformly at random. We are interested in estimating statistics of $\mathcal{W}_i$.
- For each instance, estimate statistics of $\mathcal{W}_i$ by simulation as follows. Draw 20,000 (40,000) samples when the service times are normally (Pareto-) distributed from the distributions of the queue populations and the service times. Generate also the permutation σ uniformly at random based on the queue population. For each sample, record the simulated clearing time of the ith queue.[19] For each instance, record the average clearing time and the 95-, 97-, and 99-percentiles of the clearing time distribution.
- For each instance, compute the robust clearing time at the ith queue using the formulation (2). The queue population uncertainty set is

$$\mathbb{P} := \left\{ n \in \mathbb{R}^M : \left| \sum_{i=1}^{\ell} n_i - \ell \mu^{\mathbb{P}} \right| \le \sigma^{\mathbb{P}} \Gamma \sqrt{\ell},\ \ell = 1, \dots M \right\}, \qquad \text{(B.1)}$$

where Γ is chosen to match the percentile of interest (for details, see Bandi et al. 2015). Note that to estimate the average clearing time, we heuristically select $\Gamma = 0.5$, which exhibits good numerical performance. The service time uncertainty set is as in Assumption 1, with $\Gamma_j^{\mathbb{X}} = \sigma_j \Gamma$ and $\alpha_j = 2$ (in the case of normally distributed services), or where $\Gamma_j^{\mathbb{X}}$ and α_j are chosen as in Section 2.1 of Bandi et al. (2015) (in the case of Pareto-distributed services). For each of the four statistics, record W_i.
- Compute the average absolute relative error as in Section 3.3 across all 100 instances.

Table B.1. Average Absolute Relative Errors (in %) of Our Estimates When Services Are Normally Distributed

	$K = M = 10$			$K = M = 20$			$K = M = 50$		
Statistic	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$
$\sigma_s = 2.5$									
Average	8.65	7.78	6.46	7.39	7.22	5.32	6.8	6.05	4.35
95-%ile	5.14	3.32	2.82	1.06	3.04	2.19	0.87	1.53	1.03
97-%ile	4.04	2.26	2.98	0.44	3.12	2.25	0.60	1.99	1.10
99-%ile	3.54	1.54	1.27	2.35	4.98	2.73	1.27	2.89	0.62
$\sigma_s = 4.0$									
Average	8.21	7.54	6.12	6.84	6.9	5.49	6.47	6.33	4.67
95-%ile	2.23	2.57	2.44	0.64	3.28	3.59	1.21	2.60	2.11
97-%ile	1.75	2.16	1.65	1.49	4.14	4.85	0.59	3.33	3.39
99-%ile	5.05	4.09	3.51	4.47	7.70	5.31	2.83	5.08	1.50

Table B.2. Average Absolute Relative Errors (in %) of Our Estimates When Services Are Pareto Distributed

	$K = M = 10$			$K = M = 20$			$K = M = 50$		
Statistic	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$	$\mu^{\mathbb{P}} = 10$	$\mu^{\mathbb{P}} = 50$	$\mu^{\mathbb{P}} = 100$
$\alpha = 1.5$									
Average	7.65	7.17	6.09	6.87	7.26	5.38	6.66	6.15	4.2
95-%ile	5.66	4.63	3.82	1.38	2.89	1.67	0.68	2.64	1.46
97-%ile	4.89	2.54	7.49	0.87	2.44	2.11	0.84	1.47	0.98
99-%ile	2.36	1.62	4.99	0.96	3.08	0.97	0.40	2.54	1.66
$\alpha = 1.7$									
Average	8.24	7.50	6.42	6.47	7.01	5.33	6.74	6.50	4.49
95-%ile	4.84	5.75	5.64	2.16	2.09	2.50	1.71	1.94	1.78
95-%ile	1.56	2.86	5.28	1.00	4.65	4.08	1.03	2.82	2.91
99-%ile	3.69	5.13	7.25	4.10	6.49	8.99	1.27	4.00	2.52

Our results are summarized in Tables B.1 and B.2 for the cases of normally distributed and Pareto-distributed services, respectively. The tables showcase that, across all experiments, the average absolute relative errors of our approach are under 9%.

B.2. Unknown Queue Population Distribution

We now investigate the setting when the true queue population distribution is not perfectly known and instead a different distribution is assumed. In this case, the simulation approach fails to deliver accurate estimates for the clearing time of a queue. We thus benchmark the estimates obtained using both our approach and simulation against that of an oracle that knows the true distribution. Across all of our experiments, the service times are assumed to be normally distributed with mean $1/\mu_j = 1$ and standard deviation σ_j equal to either 25% or 40% with both parameters perfectly known. The following procedure underlies all of our experiments:

- Let $K = M = 20$. Select the mean $\mu^{\mathbb{P}}$ of each queue's population distribution. The populations of all queues are independent and either normally distributed with standard deviation $\sigma^{\mathbb{P}}$ or Pareto distributed with parameter α. Only the means of the (otherwise unknown) queue population distributions are known. Also select the value of σ_j uniformly at random. Holding these parameters fixed, generate 100 instances of the problem by constructing server eligibility sets $\mathbb{S}$ at random. For each instance, select a queue i

Table B.3. Average Absolute Relative Errors of Both Our Estimates and Simulation Estimates for the Average Wait Time When the Queue Population Distribution Assumed Differs from the Actual Distribution for the Case When the Average Queue Population Is $\mu^{\mathbb{P}} = 5$

Queue population distribution			
True	Assumed	W_i $(\Gamma = 0.5)$ (%)	Simulation (%)
Normal(5, 10)	Normal(5, 5)	13.23	11.10
	Normal(5, 10)	11.69	0.00
	Normal(5, 15)	12.38	20.89
	Normal(5, 20)	13.92	23.18
	Exponential(5)	12.45	21.59
Pareto(5, 1.5)	Normal(5, 5)	11.11	21.89
	Normal(5, 10)	11.55	21.34
	Normal(5, 15)	12.38	17.87
	Normal(5, 20)	12.37	16.75
	Exponential(5)	12.11	20.92
	Pareto(5, 1.7)	11.16	19.67
	Pareto(5, 1.3)	12.02	33.85
Pareto(5, 1.7)	Normal(5, 5)	13.70	27.98
	Normal(5, 10)	14.94	23.52
	Normal(5, 15)	15.07	21.24
	Normal(5, 20)	13.67	20.14
	Exponential(5)	15.94	24.26
	Pareto(5, 1.5)	14.20	21.13
	Pareto(5, 1.3)	14.88	31.99
Avg. abs. relative error across all instances		13.09	21.02

Table B.4. Average Absolute Relative Errors of Both Our Estimates and Simulation Estimates for the Average Wait Time When the Queue Population Distribution Assumed Differs from the Actual Distribution for the Case When the Average Queue Population Is $\mu^{\mathbb{P}} = 100$

Queue population distribution			
True	Assumed	W_i $(\Gamma = 0.5)$ (%)	Simulation (%)
Normal(100, 50)	Normal(100, 25)	7.88	6.73
	Normal(100, 50)	8.18	0.00
	Normal(100, 75)	10.89	13.89
	Normal(100, 100)	9.08	19.09
	Exponential(100)	8.14	17.36
Pareto(100, 1.5)	Normal(100, 25)	8.65	14.74
	Normal(100, 50)	9.68	12.66
	Normal(100, 75)	8.23	14.73
	Normal(100, 100)	8.95	12.59
	Exponential(100)	7.90	11.82
	Pareto(100, 1.7)	9.57	11.29
	Pareto(100, 1.3)	7.07	22.51
Pareto(100, 1.7)	Normal(100, 25)	10.47	19.06
	Normal(100, 50)	10.06	16.14
	Normal(100, 75)	10.48	17.93
	Normal(100, 100)	8.60	14.88
	Exponential(100)	8.68	19.78
	Pareto(100, 1.5)	12.29	15.83
	Pareto(100, 1.3)	12.70	24.24
Avg. abs. relative error across all instances		9.34	15.01

Table B.5. Average Absolute Relative Errors of Both Our Estimates and Simulation Estimates for the Average Wait Time When the Queue Population Distribution Assumed Differs from the Actual Distribution for the Case When the Average Queue Population Is $\mu^{\mathbb{P}} = 500$

Queue population distribution			
True	Assumed	W_i ($\Gamma = 0.5$) (%)	Simulation (%)
Normal(500, 200)	Normal(500, 150)	6.60	5.26
	Normal(500, 200)	6.52	0.00
	Normal(500, 350)	7.55	12.34
	Normal(500, 500)	6.50	15.88
	Exponential(500)	7.41	14.78
Pareto(500, 1.5)	Normal(500, 150)	8.26	12.58
	Normal(500, 200)	7.22	9.05
	Normal(500, 350)	7.14	11.98
	Normal(500, 500)	7.06	10.05
	Exponential(500)	6.67	10.01
	Pareto(500, 1.7)	8.55	10.07
	Pareto(500, 1.3)	5.76	16.94
Pareto(500, 1.7)	Normal(500, 150)	8.26	15.26
	Normal(500, 200)	7.50	11.75
	Normal(500, 350)	9.34	15.41
	Normal(500, 500)	8.29	13.23
	Exponential(500)	6.82	15.23
	Pareto(500, 1.5)	9.01	11.22
	Pareto(500, 1.3)	11.13	19.42
Avg. abs. relative error across all instances		7.66	12.13

randomly. We are interested in estimating the average clearing time of the ith queue, $\mathcal{W}_i$. Select an assumed distribution for the queue population with mean $\mu^{\mathbb{P}}$. This can be either normal, Pareto, or exponential.

- For each instance, use simulation to compute the true expected clearing time of the ith queue using a procedure that parallels that from Section B.1. Note that in reality, this estimate would not be possible to obtain since the queue population distributions are unknown.
- Estimate the average clearing time of the ith queue under the assumed distribution using both simulation and our approach, in the exact same fashion as described in Section B.1.
- Compute the average absolute relative error of both approaches relative to the true value returned by the oracle across all 100 instances.

Our results are summarized in Tables B.3–B.5 for $\mu^{\mathbb{P}} = 5$, 100, and 500, respectively. We observe that the average absolute relative error of the simulation approach is consistently greater by a factor of over 1.5 relative to our approach, and this independently of the value of $\mu^{\mathbb{P}}$. Moreover, we observe that our method converges as $\mu^{\mathbb{P}}$ increases, consistent with the CLT asymptotic behavior.

B.3. Computation Times

We conclude with a summary of the computation times taken by our approach.[20] We computed the average solver times taken by our method over 100 randomly generated instances, for a varying number of classes and an average queue population $\mu^{\mathbb{P}} = 50$, as in Section B.1. We observe that for instances even as large as $K = M = 500$—i.e., instances involving an average number of $50 \times 500 = 25{,}000$ customers—the average solver times were under two minutes (see Table B.6).

Table B.6. Computation Times for Different Problem Sizes

$K = M =$	10	20	50	100	500
Solver time (seconds)	0.42	0.93	17.2	39.6	152.4

Appendix C. Numerical Experiments on Synthetic Instances of HMCMS Systems

C.1. Computation Times

To evaluate the required computation times of the MIP (3) and the SOCP (4) (for $\alpha_j = 2$), we used both formulations to compute W_K in randomly generated instances of HMCMS systems. For benchmark purposes, we computed W_K using also the general MIP formulation (2). The instances were generated as follows.

- Select the number of classes (and servers) $K (= M)$ among the values $\{10, 20, 50, 100, 200, 500\}$.
- Select also the means $\{\hat{n}_i\}_{i=1,\dots,K}$ of each queue's population distribution among the values $\{10, 20, 50, 100, 200, 500\}$.
- Construct the uncertainty sets $\mathbb{P}$ (as in (5)) with the parameters $\{\hat{n}_i\}_{i=1,\dots,K}$ and $\Gamma^{\mathbb{P}} = 2/\sqrt{K}$. This gives rise to on average a total of $K \cdot \hat{n}_i$ customers in the system.
- Holding these parameters fixed, generate 100 instances of the problem by randomly varying the service rates μ_j. For each instance, solve the optimization problems (2)–(4) while measuring the solver times.

Our results are included in Tables C.1–C.3.

C.2. Accuracy of Heuristic Approach

To evaluate the accuracy of the SOCP (4), we used it to compute $\hat{W}_K$ for randomly generated instances of HMCMS systems, and measured the approximation error compared with W_K. Our approach was:

- For various lower and upper bounds on queue populations, $\underline{p}$ and $\bar{p}$, respectively, generate 1,000 instances as follows.
- Let $K = M = 5$. Select the ith class population n_i randomly between $[\underline{p}, \bar{p}]$. Let $\mathbb{P} = \{n\}$. Select arrival rates μ_j randomly between $[0.1, 1.1]$ and Γ^{x} randomly between $[0, 1]$.
- For each instance, solve SOCP (4) to compute $\hat{W}_K$, and similarly MIP (3) to compute W_K.

Table C.1. Average Computation Times (in Seconds) of MIP (2) for HMCMS Systems with Varying Size of the System (K) and Number of Customers ($\hat{n}$)

	$\hat{n}$					
K	10	20	50	100	200	500
10	1.03	1.65	26.67	110.33	261.49	470.22
20	1.18	8.22	29.23	237.16	315.48	574.41
50	7.1	48.28	101.35	324.94	414.96	580.22
100	20.48	92.52	156.56	380.56	692.65	916.09
200	94.53	132.92	258.73	447.55	2,348.72	2,755.79
500	135.92	268.25	483.92	985.09	2244	4,656.11

Table C.2. Average Computation Times (in Seconds) of MIP (3) for HMCMS Systems with Varying Size of the System (K) and Number of Customers ($\hat{n}$)

	$\hat{n}$					
K	10	20	50	100	200	500
10	0.84	1.11	9.96	40.46	60.45	115.42
20	1.09	5.37	15.26	70.28	84.72	135.05
50	5.42	24.56	55.37	104.89	129.79	149.97
100	19.55	54.73	69.93	125.34	165.13	199.89
200	64.86	85.34	120.23	189.53	540.19	650.37
500	94.84	134.77	179.89	399.95	649.54	1,149.74

Table C.3. Average Computation Times (in Seconds) of SOCP (4) for HMCMS Systems with Varying Size of the System (K) and Number of Customers ($\hat{n}$)

	$\hat{n}$					
K	10	20	50	100	200	500
10	0.83	0.89	0.98	1.18	1.92	2.45
20	0.86	0.93	1.19	2.12	2.29	2.62
50	1.17	1.06	1.95	2.59	3.29	3.71
100	1.72	3.37	3.08	3.97	7.1	10.25
200	1.83	5.22	8.53	15.39	19.23	42.64
500	3.38	8.81	12.28	19.11	44.43	74.32

Table C.4. Average Relative Approximation Error of Our SOCP Heuristic (4) for HMCMS Systems with Varying Number of Customers

Lower and upper bounds on queue populations $[\underline{p}, \bar{p}]$	Avg. relative error $((\hat{W}_K - W_K)/W_K) \times 100\%$ (%)
$[5, 10]$ customers	1.9
$[15, 30]$ customers	0.85
$[25, 50]$ customers	0.5
$[75, 100]$ customers	0.25
$[200, 300]$ customers	0.08

- Compute the average approximation error across all 1,000 instances.

Table C.4 reports our results. Evidently, our heuristic is almost exact and becomes tighter as $\underline{p}$ and $\bar{p}$ increase—i.e., as population sizes grow.

Appendix D. Estimating Kidney Patients' Preferences

We outline the procedure we followed in our case study in Section 5 to estimate q_i, the probability of a random wait-listed patient being an i-patient—i.e., being willing to accept a kidney if and only if it is of quality i or higher, for all $i = 1, \ldots, K$. For simplicity, we assume here that all patients are available—i.e., $q_0 = 0$. Put differently, we discuss how to calculate the probability of a patient being in class i, conditional on being available. The unconditional probabilities can be readily retrieved by scaling the conditional ones by $1 - q_0$.

At a high level, our approach is to estimate the probabilities with the ones that maximize the likelihood of the recorded offer decisions in the UNOS data set. In particular, for all $k = 1, \ldots, K$, let $\mathcal{A}^k$ and $\mathcal{Q}^k$ be indicator random variables such that

$$\mathcal{A}^k = \begin{cases} 1 & \text{if the patient is willing to accept} \\ & \quad \text{kidneys of quality } k, \\ 0 & \text{otherwise;} \end{cases}$$

$$\mathcal{Q}^k = \begin{cases} 1 & \text{if the patient is a } k\text{-patient,} \\ 0 & \text{otherwise.} \end{cases}$$

By definition,

$$\mathbf{P}(\mathcal{Q}^i = 1) = q_i \quad \text{and} \quad \mathbf{P}(\mathcal{A}^j = 1 | \mathcal{Q}^i = 1) = \begin{cases} 1 & \text{if } j \le i, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i, j = 1, \ldots, K$. Thus,

$$\begin{aligned} \mathbf{P}(\mathcal{A}^j = 1) &= \sum_{i=1}^{K} \mathbf{P}(\mathcal{A}^j = 1 \mid \mathcal{Q}^i = 1) \mathbf{P}(\mathcal{Q}^i = 1) \\ &= \sum_{i=j}^{K} \mathbf{P}(\mathcal{Q}^i = 1) = \sum_{i=j}^{K} q_j, \quad j = 1, \ldots, K. \end{aligned}$$

Let a_i (r_i), $i = 1, \ldots, K$, denote the records in the UNOS data set of a kidney of quality i being accepted (rejected) because of quality. The likelihood of observing a_i (r_i) accept (reject) decisions for kidneys of quality i can be readily expressed as

$$\sum_{i=1}^{K} a_i \log\left(\sum_{j=i}^{K} q_j\right) + r_i \log\left(1 - \sum_{j=i}^{K} q_j\right),$$

for all $i = 1, \ldots, K$. Note that in line with the literature, we assumed that decisions are independent of each other and are solely driven by kidney quality—see, e.g., Zenios (2005). Then, the maximum likelihood probabilities can be obtained by solving the following convex optimization problem in the variables $q_1, \ldots, q_K$:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{K} a_i \log\left(\sum_{j=i}^{K} q_j\right) + r_i \log\left(1 - \sum_{j=i}^{K} q_j\right) \\ \text{subject to} \quad & \sum_{i=1}^{K} q_i = 1, \\ & q_i \ge 0 \quad i = 1, \ldots, K. \end{aligned}$$

Appendix E. Hybrid Priority Systems

In this section, we study HMCMS systems where some servers follow CP and others follow FCFS. We refer to such priority rules as *hybrid* (HP). As with our analysis of class-priority systems, we again focus our discussion on a specific model that pertains to KAS because of space considerations—more general cases can be tackled in a similar fashion.

Consider an HMCMS system for which we are interested in estimating the clearing time of the Kth queue, as in Section 4.1. There is an additional class, indexed by $i = 0$, who seek service from the 1st server only—i.e., the one providing the highest service quality. That is, $\mathbb{S}(0) = \{1\}$ and $\mathbb{Q}(1) = \{0, 1, \ldots, K\}$. Server 1 prioritizes 0-customers over all other customers. All other model specifications are as in Section 4. In particular, all servers but the first one follow FCFS.

This model adequately captures the dynamics under the new KAS. Specifically, patients with an EPTS score in the top-20% range can be classified in the 0th class. Consequently, they would receive priority for top-quality organs (procured by server 1) over all other patients.[21]

In this context, it can be readily seen that only arrivals of 0-customers affect the Kth queue's clearing time, and are thus the only arrivals we model. We refrain from formalizing further model dynamics of this hybrid HMCMS system, as they closely resemble the dynamics of FCFS and CP systems we outlined in Sections 2 and 6.1. We also use uncertainty models, notation and solution methodology that are immediate extensions of our approach so far. For instance, we denote the (robust) clearing time we are interested in with (W_K^{HP}) $\mathcal{W}_K^{\mathrm{HP}}$.

In this context, one can readily extend our analysis to show that calculating W_K^{HP} is $\mathcal{NP}$-hard and the following monotonicity result.

Lemma 3. *For a hierarchical MCMS system under HP, the clearing time $\mathcal{W}_K^{\mathrm{HP}}$ is increasing in the service times $x_1, \dots, x_K$ and decreasing in the arrival times a_0.*

Using Lemma 3, we fix the completion and arrival times to their worst-case values as in Section 6.1. The following MIP, which builds on problem (3), allows us to compute W_K^{HP}.

$$\text{maximize} \quad \sum_{\ell=2,\dots,\bar{\ell}} c^{\ell}(f^{\ell-1} - f^{\ell}) \tag{E.1a}$$

$$\text{subject to} \quad \text{constraints (3a)–(3i)} \tag{E.1b}$$

$$\sum_{(j,\omega):\, c_1^{\omega} \le c^{\ell}} y_{01}^{\omega} \le n_0 + v_0^{\ell} - f_0^{\ell}, \quad \ell = 1, \dots, \bar{\ell}; \tag{E.1c}$$

$$y_{i1}^{\omega} \le 1 - f_0^{\ell}, \quad i \ge 1,\ \omega\colon c_1^{\omega} = c^{\ell},\ \ell = 1, \dots, \bar{\ell}; \tag{E.1d}$$

$$f_0^{\ell} \in \{0,1\},\ \ell = 1, \dots, \bar{\ell}. \tag{E.1e}$$

Theorem 6. *For the hierarchical MCMS system under hybrid priority defined above, the optimal value of the MIP (E.1) is equal to W_K^{HP}.*

Loosely speaking, MIP (E.1) builds on formulation (3) to capture the FCFS dynamics of the original system, as reflected in the common constraints (3a)–(3i). MIP (E.1) then borrows from (7) the CP dynamics that pertain to the 0th class, as reflected in the additional constraints (E.1c)–(E.1d). In particular, variables f_0^{ℓ} indicate whether class 0 is filled or has cleared by time c^{ℓ}. Constraint (E.1c) is then an arrivals-adjusted capacity constraint for the 0th class, similar to (7d). Constraint (E.1d) enforces the CP priority: at any c^{ℓ}, if the 0th class is filled, the 1st server cannot serve any lower priority $i > 0$ class—i.e., $y_{i1}^{\omega} \le 1 - f_0^{\ell} = 0$ (similar to constraint (7g).

In summary, our treatment in this section demonstrated the flexibility of our modeling framework to tackle multiclass multiserver queuing systems under priority rules different than FCFS that are also potentially open. While we limited our exposition to the particular hierarchical service systems for brevity, our approach is still applicable in the general case.

Appendix F. Proofs

We present the proofs of the main results in the order in which they appear in our paper.

Proof of Proposition 1. Consider the decision problem associated with the optimization problem (1), where we query whether its optimal value is greater than or equal to some value V. Let Π denote this decision problem. We will show that the problem Partition (Garey and Johnson 1979), which is known to be $\mathcal{NP}$-hard, transforms to Π. That is, given an instance I_{P} of Partition, we will show how to construct an instance I_{Π} of Π in polynomial time, such that I_{P} is a Yes instance of Partition if and only if I_{Π} is a Yes instance of Π.

To introduce some notation, we define the decision problem.

Partition:
Instance: A set of k positive integers $\mathbb{A} = \{a_1, \dots, a_k\}$, with $\sum_{\ell=1}^{k} a_{\ell} = 2B$, $B \in \mathbb{N}$.
Query: Is there a subset $\mathbb{A}_1 \subset \mathbb{A}$ such that $\sum_{\ell \in \mathbb{A}_1} a_{\ell} = \sum_{\ell \in \mathbb{A} \setminus \mathbb{A}_1} a_{\ell} = B$?

We construct an instance I_{Π} of Π as follows:

(i) $K = 2$, $M = 2$, with $\mathbb{S}(1) = \{1\}$, $\mathbb{S}(2) = \{1, 2\}$.
(ii) $i = 2$.
(iii) $\mathbb{P} \cap \mathbb{N}^2 = \{n\colon\ n_1 = y^{\top}a, n_2 = 2B - y^{\top}a, n_1 \ge B, y \in \{0,1\}^k\}$.
(iv) $\mu_1 = \mu_2 = 1$, $\Gamma_1^{\mathbb{X}} = \Gamma_2^{\mathbb{X}} = 0$, $\bar{\ell}_1 = \bar{\ell}_2 = k$.
(v) $V = B$.

For the constructed instance, note that there are always $2B$ customers in the system, split between the two classes, with class 1 having at least B customers. All service times are equal to one. For the worst-case clearing time W_2, we can take without loss the service priority of the 1-customers to be higher than all of 2-customers. This ensures that server 1 does not serve any 2-customer. Therefore, in the worst case, we have that $W_2 = n_2$.

Suppose now that I_{P} is a Yes instance of Partition. Then, let $y_{\ell} = \mathcal{I}(\ell \in \mathbb{A}_1)$, for all $\ell = 1, \dots, k$. This value of y yields a population vector $n_1 = n_2 = B$, and therefore I_{Π} is a Yes instance of Π since $W_2 = B \ge V$. Conversely, if I_{Π} is a Yes instance of Π, we conclude that $W_2 = n_2 = B$ for some population vector n, such that $n_1 = n_2 = B$. Let $y \in \{0,1\}^k$ the corresponding vector that generates n. By letting $\mathbb{A}_1 = \{\ell\colon\ y_{\ell} = 1\}$, we get that

$$\sum_{\ell \in \mathbb{A}_1} a_{\ell} = n_1 = B,$$

and I_{P} is a Yes instance of Partition. □

Proof of Theorem 1. We proceed in two steps. First, we show that W_i is equal to the optimal value of the following optimization problem with variables w, $n \in \mathbb{R}^K$, $q \in \mathbb{R}^{|\mathbb{Q}(1)|\bar{\ell}_1 + \cdots + |\mathbb{Q}(M)|\bar{\ell}_M}$, and $c \in \mathbb{R}^{\Sigma_{j=1}^{M} \bar{\ell}_j}$.

$$\text{maximize} \quad w_i \tag{F.1a}$$

$$\text{subject to} \quad q_j^{\ell} \in \mathbb{Q}(j) \cup \{K+1\}, \quad \ell = 1, \dots, \bar{\ell}_j,\ j = 1, \dots, M; \tag{F.1b}$$

$$q_j^{\ell} \in \mathbb{Q}(j), \quad \ell = 1, \dots, \bar{\ell}_j,\ j = 1, \dots, M\colon c_j^{\ell} < w_k \ \text{for some } k \in \mathbb{Q}(j); \tag{F.1c}$$

$$\sum_{\substack{\ell=1,\dots,\bar{\ell}_j \\ j \in \mathbb{S}(k)}} \mathcal{I}(q_j^{\ell} = k) = n_k, \quad k = 1, \dots, K; \tag{F.1d}$$

$$w_k = \max\{c_j^{\ell}\colon q_j^{\ell} = k \text{ or } \ell = 0, j \in \mathbb{S}(k), \ell = 0, \dots, \bar{\ell}_j\}, \quad k = 1, \dots K; \tag{F.1e}$$

$$c_j \in \mathbb{C}_j, \quad j = 1, \dots, M; \tag{F.1f}$$

$$n \in \mathbb{P} \cap \mathbb{N}^K, \tag{F.1g}$$

where we use the convention that $c_j^0 = 0$ for all $j \in \{1, \dots, M\}$. Problem (F.1) admits a very intuitive interpretation. The variables q_j^ℓ model the queue the jth server assigns its ℓth service, as per constraint (F.1b). Service is assigned to the fictitious queue $K+1$ if no eligible customer is available for service. Constraint (F.1c) captures the fact that if at time c_j^ℓ, there exists a nonempty queue compatible with server j, then the ℓth service from the jth server cannot be assigned to the fictitious queue. Constraint (F.1d) requires that all customers from all queues are served, while constraint (F.1e) corresponds to the definition of the completion time of a queue, with the completion time being equal to zero if no customers were waiting.

Proposition 3. *The optimal values of problems* (1) *and* (F.1) *are finite and equal to each other. Moreover, for every optimal solution* (n, σ, x) *to* (1), *there exists an optimal solution* (w, n, q, c) *to* (F.1) *such that* $c_j^\ell = x_j^1 + \cdots + x_j^\ell$, $j = 1, \dots, M$, $\ell = 1, \dots, \bar{\ell}_j$, *and vice versa.*

Second, we show that problems (2) and (F.1) have the same optimal value.

Proposition 4. *The optimal values of problems* (2) *and* (F.1) *are equal to each other. Moreover, for every optimal solution to* (F.1), *there exists an optimal solution to* (2) *such that the optimal vectors of completion times coincide.* □

Proof of Proposition 2. Follows directly from the proof of Proposition 1. □

Proof of Lemma 1. Recall that in a hierarchical MCMS, $\mathbb{Q}(j) = \{j, \dots, K\}$ for all $j \in \{1, \dots, K\}$ and $\mathbb{S}(k) = \{1, \dots, k\}$ for all $k \in \{1, \dots, K\}$. Proposition 3 implies that W_K is equal to the optimal value of (F.1) with $i := K$. We show that given any feasible solution (w, n, q, c) to (F.1) and any sequence of service times $\tilde{c}$ such that $\tilde{c}_j \in \mathbb{C}_j$ and $\tilde{c}_j^\ell \geq c_j^\ell$ for all $j \in \{1, \dots, K\}$ and $\ell \in \{1, \dots, \bar{\ell}_j\}$, there exists a solution $(\tilde{w}, n, \tilde{q}, \tilde{c})$ feasible in (F.1) and such that $\tilde{w}_K \geq w_K$. This will enable us to conclude that there exists an optimal solution to (F.1) in which the completion times all attain their maximum values. The proof of this lemma will then readily follow from Proposition 3.

Let (w, n, q, c) be feasible in (F.1), and let $\tilde{c}$ such that $\tilde{c}_j \in \mathbb{C}_j$ and $\tilde{c}_j^\ell \geq c_j^\ell$ for all $j \in \{1, \dots, K\}$ and $\ell \in \{1, \dots, \bar{\ell}_j\}$. Also, define an assignment r and a population $\hat{n}$ as follows:

$$r_j^\ell := \begin{cases} q_j^\ell & \text{if } \tilde{c}_j^\ell < w_K, \\ K+1 & \text{else;} \end{cases}$$

$$\hat{n}_k := n_k - \sum_{\substack{\ell=1,\dots,\bar{\ell}_j \\ j=1,\dots,k}} \mathcal{I}(r_j^\ell = k),$$

for all $j \in \{1, \dots, K\}$, $\ell \in \{1, \dots, \bar{\ell}_j\}$, and $k \in \{1, \dots, K\}$. Note that $\hat{n} \geq 0$, and in particular $\hat{n}_k > 0$ for all $k \in \{1, \dots, K\}$ such that $w_k \geq w_K$. To see the latter, fix $k \in \{1, \dots, K\}$ such that $w_k \geq w_K$. Then, $\hat{n}_k \leq 0$ would imply that more than n_k k-customers are served under assignment r. Since under r, customers are served only at times before w_K according to q (and servers remain idle afterward), this would imply that more than n_k k-customers are served under assignment q before w_K, a contradiction since the earliest time at which the n_kth k-customer is served is w_K.

Let $\underline{\ell}_j$ be the number of customers served by the jth server under r—i.e.,

$$\underline{\ell}_j := \max\{\ell\colon r_j^\ell < K+1\}, \quad j = 1, \dots, K.$$

Consequently, the times their service started has to be less than w_K (by the definition of r). Thus, for all $j \in \{1, \dots, K\}$ and $\ell \in \{1, \dots, \underline{\ell}_j\}$, it holds that

$$\tilde{c}_j^\ell \begin{cases} < w_K & \text{if } \ell \leq \underline{\ell}_j, \\ \geq w_K & \text{else.} \end{cases}$$

Consider now a new instance of problem (F.1) with identical service system layout, but where the queue population uncertainty set is given by the singleton $\{\hat{n}\}$ and where the uncertainty set for the server completion times is given by the singleton $\{\hat{c}\}$, where $\hat{c}$ is defined through $\hat{c}_j^\ell = \tilde{c}_j^{\ell + \underline{\ell}_j}$, $j \in \{1, \dots, K\}$, $\ell \in \{1, \dots, \bar{\ell}_j - \underline{\ell}_j\}$. Let $(\hat{w}, \hat{q})$ be such that $(\hat{w}, \hat{n}, \hat{q}, \hat{c})$ is feasible in the associated instance of problem (F.1). Next, for $j \in \{1, \dots, K\}$ and $\ell \in \{1, \dots, \bar{\ell}_j - \underline{\ell}_j\}$, define

$$\tilde{q}_j^\ell := \begin{cases} q_j^\ell & \text{if } \ell \leq \underline{\ell}_j, \\ \hat{q}_j^{\ell - \underline{\ell}_j} & \text{else;} \end{cases}$$

$$\tilde{w}_k := \begin{cases} w_k & \text{if } w_k < w_K, \\ \hat{w}_k & \text{else.} \end{cases}$$

We first argue that $w_K \leq \tilde{w}_K$. The definition of $\hat{n}$ implies that $\hat{n}_K > 0$ and therefore feasibility of $(\hat{w}, \hat{n}, \hat{q}, \hat{c}, \hat{w})$ in the instance of (F.1) implies $\hat{w}_K \in \hat{c}$. But, for all $j \in \{1, \dots, K\}$ and $\ell \in \{1, \dots, \bar{\ell}_j - \underline{\ell}_j\}$, $\hat{c}_j^\ell = \tilde{c}_j^{\ell + \underline{\ell}_j} \geq w_K$. Hence, $w_K \leq \hat{w}_K = \tilde{w}_K$.

The final step is to show that $(\tilde{w}, n, \tilde{q}, \tilde{c})$ is feasible in problem (F.1). Constraint (F.1b) is trivially satisfied. For (F.1c), fix $j \in \{1, \dots, K\}$. Then:

- For $\ell \leq \underline{\ell}_j$, we have $c_j^\ell \leq \tilde{c}_j^\ell < w_K \leq \tilde{w}_K$. Since $K \in \mathbb{Q}(j)$, feasibility of (w, n, q, c) in (F.1) combined with $c_j^\ell < w_K$ imply that $q_j^\ell \in \mathbb{Q}(j)$. The definition of $\tilde{q}$ then yields $\tilde{q}_j^\ell = q_j^\ell \in \mathbb{Q}(j)$, and constraint (F.1c) is satisfied in this case;
- For $\ell > \underline{\ell}_j$, if $\exists k \in \mathbb{Q}(j)$ such that $\tilde{c}_j^\ell < \tilde{w}_k$, then the definition of $\tilde{c}$ implies that $w_K \leq \tilde{c}_j^\ell < w_k$ and therefore it follows from the definition of $\tilde{w}$ that $\tilde{w}_k = \hat{w}_k$. Therefore, $\hat{c}_j^{\ell - \underline{\ell}_j} = \tilde{c}_j^\ell < \tilde{w}_k = \hat{w}_k$. The feasibility of $(\hat{w}, \hat{n}, \hat{q}, \hat{c})$ in its corresponding instance of (F.1) implies $\hat{q}_j^{\ell - \underline{\ell}_j} \in \mathbb{Q}(j)$. The definition of $\tilde{q}$ yields $\tilde{q}_j^\ell = \hat{q}_j^{\ell - \underline{\ell}_j} \in \mathbb{Q}(j)$, and constraint (F.1c) holds.

As the choice of j was arbitrary, constraint (F.1c) is satisfied. For (F.1d), we have that

$$\begin{aligned} \sum_{\substack{\ell=1,\dots,\bar{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(\tilde{q}_j^\ell = k) &= \sum_{\substack{\ell=1,\dots,\underline{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(\tilde{q}_j^\ell = k) + \sum_{\substack{\ell=\underline{\ell}_j+1,\dots,\bar{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(\tilde{q}_j^\ell = k) \\ &= \sum_{\substack{\ell=1,\dots,\underline{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(q_j^\ell = k) + \sum_{\substack{\ell=\underline{\ell}_j+1,\dots,\bar{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(\hat{q}_j^{\ell - \underline{\ell}_j} = k) \\ &= n_k - \hat{n}_k + \sum_{\substack{\ell=\underline{\ell}_j+1,\dots,\bar{\ell}_j \\ j=1,\dots,K}} \mathcal{I}(\hat{q}_j^{\ell - \underline{\ell}_j} = k) \\ &\qquad\qquad [\text{by definitions of } \hat{n}, r, \underline{\ell}_j] \\ &= n_k - \hat{n}_k + \hat{n}_k = n_k \\ &\qquad\qquad [\text{by feasibility of } (\hat{q}, \hat{c}, \hat{w})]. \end{aligned}$$

Finally, it can be readily checked that $(\tilde{w}, n, \tilde{q}, \tilde{c})$ satisfies constraint (F.1e) by the definition of $\tilde{q}$ and the fact that $\hat{n}_k > 0$ for all $k \in \{1, \ldots, K\}$ such that $w_k \geq w_K$. □

Proof of Theorem 2. Recall that, in the context of hierarchical MCMS, $\mathbb{Q}(j) = \{j, \ldots, K\}$ for all $j \in \{1, \ldots, K\}$ and $\mathbb{S}(k) = \{1, \ldots, k\}$ for all $k \in \{1, \ldots, K\}$. Theorem 1 implies that W_K is equal to the optimal value of problem (2) with $i := K$. It thus suffices to show that the optimal values of problems (2) and (3) are equal in the present setting.

Let (w, n, y, f, c) be an optimal solution to problem (2) such that the completion times are equal to their worst-case values. Existence of such a solution is guaranteed by Lemma 1 and Propositions 3 and 4. We first argue that $\exists\, j^\star, \ell^\star, t^\star$ such that

$$w_K = c_{j^\star}^{\ell^\star} = c^{t^\star} \quad \text{and} \quad f_{Kj}^{\ell} = \begin{cases} 1 & \text{if } c_j^{\ell} < w_K, \\ 0 & \text{if } \ell = \ell^\star \text{ and } j = j^\star. \end{cases}$$

To see this, note that if $f_{Kj}^{\ell} = 1$ for all j and ℓ, then by (2e), w_K can take a value that is strictly bigger than $\bar{\zeta}$ (since all of the elements of c are positive), a contradiction. Let then $(j^\star, \ell^\star) \in \arg\min\{c_j^{\ell} \colon f_{Kj}^{\ell} = 0\}$. Then, by optimality of (w, n, y, f, c), constraint (2e) is binding for $j^\star$ and $\ell^\star$, and our claim follows. Define the variables $\hat{f} \in \mathbb{R}^{\bar{\ell}}$ and $\hat{n} \in \mathbb{R}^K$ such that for $\ell \in \{1, \ldots, \bar{\ell}\}$,

$$\hat{f}^{\ell} := \begin{cases} 1 & \text{if } \ell < t^\star, \\ 0 & \text{else;} \end{cases}$$
$$\hat{n} := n + \mathrm{e}_K.$$

We now demonstrate that $(y, \hat{n}, \hat{f})$ is feasible in problem (3), and produces an objective value (3a) equal to w_K—i.e., the optimal value of problem (2). Constraints (3b) and (3d) follow directly from (2b) and (2c), respectively. For (3c), note that

$$\sum_{(j,\omega)\colon c_j^{\omega} \leq c^{\ell}} y_{Kj}^{\omega} \leq \sum_{\substack{\omega=1,\ldots,\bar{\ell}_j \\ j=1,\ldots,k}} y_{Kj}^{\omega} \leq n_K = \hat{n}_K - 1 \leq \hat{n}_K - \hat{f}^{\ell},$$

where the second inequality follows from (2c). Constraint (3e) is trivially satisfied for $\ell \geq t^\star$. For any $\ell < t^\star$, let (j, ω) be such that $c_j^{\omega} = c^{\ell}$. Constraint (3e) then becomes $\sum_{k'=j,\ldots,K} y_{k'j}^{\omega} \geq 1$, which follows from (2d) for $k = K$ and (j, ω). Constraints (3f)–(3i) are readily satisfied. Finally, note that the objective value attained by $(y, \hat{n}, \hat{f})$ in (3) is given by $c^{t^\star}(\hat{f}^{t^\star-1} - \hat{f}^{t^\star}) = w_K$ and thus the optimal value of (3) is greater or equal to W_K.

To complete the proof, let (y, n, f) be an optimal solution to problem (3). Using a similar argument as above, $\exists\, t^\star$ such that $f^{\ell} = 1$ for $\ell \in \{1, \ldots, \bar{\ell}\}$, $\ell < t^\star$, and $f^{\ell} = 0$ else. Consequently, the optimal value of (3) is equal to $c^{t^\star}$. Define the variables $\tilde{y}, \tilde{f} \in \mathbb{R}^{K\bar{\ell}_1 + (K-1)\bar{\ell}_2 + \cdots + \bar{\ell}_K}$ such that for $j \in \{1, \ldots, M\}$, $\ell \in \{1, \ldots, \bar{\ell}_j\}$, and $k \in \mathbb{Q}(j)$,

$$\tilde{y}_{kj}^{\ell} := \begin{cases} y_{kj}^{\ell} & \text{if } c_j^{\ell} < c^{t^\star}, \\ 0 & \text{else,} \end{cases}$$

$$\tilde{f}_{kj}^{\ell} := \begin{cases} 1 & \text{if } c_j^{\ell} < c^{t^\star}, \\ 0 & \text{else.} \end{cases}$$

Consider the solution $(c^{t^\star}\mathrm{e}, n - \mathrm{e}_K, \tilde{y}, \tilde{f}, c)$, which produces an objective value (2a) equal to $c^{t^\star}$—i.e., the optimal value of problem (3). We show that $(c^{t^\star}\mathrm{e}, n - \mathrm{e}_K, \tilde{y}, \tilde{f}, c)$ is feasible in (2). Constraint (2b) follows from (3b) and from $\tilde{y} \leq y$. Similarly, for $k = 1, \ldots, K-1$, constraint (2c) follows from (3d). For $k = K$, we have

$$\begin{aligned} \sum_{\substack{\ell=1,\ldots,\bar{\ell}_j \\ j=1,\ldots,K}} \tilde{y}_{Kj}^{\ell} &= \sum_{(j,\omega)\colon c_j^{\omega} \leq c^{t^\star-1}} \tilde{y}_{Kj}^{\omega} + \sum_{(j,\omega)\colon c_j^{\omega} \geq c^{t^\star}} \tilde{y}_{Kj}^{\omega} \\ &= \sum_{(j,\omega)\colon c_j^{\omega} \leq c^{t^\star-1}} y_{Kj}^{\omega} \leq n_K - f^{t^\star-1} = n_K - 1, \end{aligned}$$

where the second equality follows from the definition of $\tilde{y}$ and the inequality from (3c). For constraint (2d), it suffices to check it for $k = K$. The constraint is trivially satisfied, unless (j, ℓ) are such that $c_j^{\ell} < c^{t^\star}$, in which case $\tilde{f}_{Kj}^{\ell} = 1$ and $\tilde{y}_{k'j}^{\ell} = y_{k'j}^{\ell}$. Let t be such that $c^t = c_j^{\ell}$. Clearly, $t < t^\star$ and thus $f^t = 1$. Constraint (2d) then follows from (3e). For constraint (2e), it again suffices to check for $k = K$. As with the previous case, for any (j, ℓ), we either have $c_j^{\ell} < c^{t^\star}$ and $\tilde{f}_{Kj}^{\ell} = 1$, or $c_j^{\ell} \geq c^{t^\star}$ and $\tilde{f}_{Kj}^{\ell} = 0$. In both cases, (2e) is trivially satisfied. Constraint (2g) is trivially valid, unless $\tilde{y}_{kj}^{\ell} = 1$—i.e., for (j, ℓ) such that $c_j^{\ell} < c^{t^\star}$. But then, the constraint becomes $c^{t^\star} \geq c_j^{\ell}$, which is true. The remaining constraints are immediate and the proof is complete. □

Proof of Theorem 3. For ease of exposition, we treat the case of $\alpha_j = 2$, $j = 1, \ldots, K$; generalizing for other values is straightforward. We introduce the following notation. Let $\mathbb{F}$ be a mapping from $\mathbb{R}^K$ to a set in $\mathbb{R}^{K+1}$ such that for all $n \in \mathbb{R}^K$

$$\mathbb{F}(n) := \left\{ (m, W) \in \mathbb{R}^{K+1} \colon W \leq \frac{m_j}{\mu_j} + \Gamma_j^{\mathbb{X}} \sqrt{m_j} \text{ and } \sum_{k=j}^{K} m_k \leq \sum_{k=j}^{K} n_k + K - j,\ j = 1, \ldots, K \right\}$$

and $\mathbb{F}_I$ be the corresponding mapping where m is integral—i.e., $\mathbb{F}_I(n) := \mathbb{F}(n) \cap \{\mathbb{N}^K \times \mathbb{R}\}$. Let h, $h_I \colon \mathbb{R}^K \to \mathbb{R}$ be such that for all $n \in \mathbb{R}^K$,

$$h(n) := \max\{W \colon (m, W) \in \mathbb{F}(n)\} \quad \text{and}$$
$$h_I(n) := \max\{W \colon (m, W) \in \mathbb{F}_I(n)\}.$$

The proof is based on the following results.

Proposition 5. *The optimal value of*

$$\begin{array}{ll} \text{maximize} & h_I(n) \\ \text{subject to} & n \in \mathbb{P} \cap \mathbb{N}^K \end{array} \tag{F.2}$$

is equal to W_K.

Proposition 6. *The optimal value of*

$$\begin{array}{ll} \text{maximize} & h(n) \\ \text{subject to} & n \in \mathbb{P} \end{array} \tag{F.3}$$

is equal to $\hat{W}_K$.

Proposition 7. *For all* $n \in \mathbb{R}^K$*, we have*

(i) $h_I(n) \leq h(n) \leq h_I(n) + \chi$.

(ii) $h_I(n) \leq h_I(m)$ *for all* $m \in \mathbb{R}^K$ *such that* $n \leq m$.

(iii) $h_I(n + \mathrm{e}) \leq h_I(n) + \chi$.

Let $n^\star \in \mathbb{P}$ be an optimal solution of problem (F.3). We then have that

$$\begin{aligned} W_K &= \max\{h_I(n)\colon n \in \mathbb{P} \cap \mathbb{N}^K\} \quad \text{[by Prop. 5]}\\ &\le \max\{h(n)\colon n \in \mathbb{P} \cap \mathbb{N}^K\} \quad \text{[by Prop. 7(i)]}\\ &\le \max\{h(n)\colon n \in \mathbb{P}\}\\ &= \hat{W}_K \quad \text{[by Prop. 6]}\\ &= h(n^\star)\\ &\le h_I(n^\star) + \chi \quad \text{[by Prop. 7(i)]}\\ &\le h_I(\lfloor n^\star \rfloor + \mathrm{e}) + \chi \quad \text{[by Prop. 7(ii)]}\\ &\le h_I(\lfloor n^\star \rfloor) + 2\chi \quad \text{[by Prop. 7(iii)]}\\ &\le W_K + 2\chi, \end{aligned}$$

where the last inequality holds since $n^\star \in \mathbb{P} \Rightarrow \lfloor n^\star \rfloor \in \mathbb{P} \cap \mathbb{N}^K$—i.e., $\lfloor n^\star \rfloor$ is feasible for problem (F.2), and Proposition 5. □

Proof of Lemma 2. We begin by defining a number of operators that will facilitate our analysis of the queue dynamics under CP. Given three ordered finite sequences $c = \{c^\ell\}_{\ell=1}^{\bar\ell}$, $a = \{a^r\}_{r=1}^{\bar r}$, and $y = \{y^m\}_{m=1}^{\bar m}$, define the operator

$$c \oplus y := \operatorname{sort}(\{c, y\}),$$

which returns the ordered finite sequence of length $(\bar\ell + \bar m)$ consisting of all elements of the concatenation of sequences c and y. Also, define the operator

$$c \to a := \{c^t\colon s^t(a,c) = 0,\ t \in \{1, \dots, \bar\ell\}\},$$

where the sequence s is given by

$$\begin{aligned} &s^0(a,c) := 0,\\ &s^t(a,c) := [s^{t-1}(a,c) - 1]^+ + z^t(a,c) - z^{t-1}(a,c)\\ &\qquad\qquad\qquad\qquad\qquad \forall\, t \in \{1, \dots, \bar\ell\}, \\ &z^t(a,c) := \max\{i \in \{0, \dots, \bar r\}\colon a^i \le c^t\}\\ &\qquad\qquad\qquad\qquad\qquad \forall\, t \in \{0, \dots, \bar\ell\}, \end{aligned} \tag{F.4}$$

with the convention that $a^0 = c^0 < 0$. These operators admit a very natural interpretation in the context of hierarchical MCMS systems under CP. The operator $c \to a$ enables us to obtain the (ordered) subset of completion times c that remain "unused" after being fed into the stream of customer arrival times a. The operator $c \oplus y$ enables us to collect (subsets of) completion times of multiple servers into a single ordered stream.

Consider a single server single class system under FCFS, where c and a collect the server completion times and the customer arrival times, respectively. We argue that the quantity $c \to a$ corresponds to the set of completion times that coincide with times when the queue was empty. For any $t \in \{0, \dots, \bar\ell\}$, the quantity $z^t(a,c) \in \{0, \dots, \bar r\}$ corresponds to the number of customers that have arrived by time c^t (note that $z^0(a,c) = 0$). Accordingly, $(z^t(a,c) - z^{t-1}(a,c)) \in \{0, \dots, \bar r\}$ represents the number of customer arrivals in the interval $(c^{t-1}, c^t]$. Interpret $s^0(a,c)$ as the number of customers waiting prior to time 0. Fix $t \in \{1, \dots, \bar\ell\}$. Suppose that $s^{t-1}(a,c) \in \mathbb{N}$ represents the number of customers waiting to be served at time c^{t-1}—i.e., the $(t-1)$th time the server completes a job. If $s^{t-1}(a,c) = 0$, the total number of customers waiting at time c^t is equal to $z^t(a,c) - z^{t-1}(a,c)$—i.e., no one was served in the interval $[c^{t-1}, c^t)$. On the other hand, if $s^{t-1}(a,c) \ge 1$, a customer is served at time c^{t-1}, and the total number of customers waiting at time c^t is given by $s^{t-1}(a,c) - 1 + z^t(a,c) - z^{t-1}(a,c)$ (a nonnegative integer). We conclude that $s^t(a,c) \in \mathbb{N}$ represents the number of people waiting to be served at time c^t for all $t \in \{0, \dots, \bar\ell\}$. Thus, for $t \in \{1, \dots, \bar\ell\}$, $s^t(a,c) = 0$ if and only if the queue is empty at time c^t, yielding the desired interpretation for $c \to a$.

We now demonstrate that $\mathcal{W}_i^{\mathrm{CP}}$ can be expressed analytically in dependence of the customer arrival times, the queue population lengths, and the server completion times using the operators introduced above.

Proposition 8. *Consider a hierarchical service system under CP with customer arrival times and server completion times given by a and c, respectively. For each $k \in \{1, \dots, i-1\}$, let*

$$\bar a_k := a_k \oplus \{\underbrace{0, \dots, 0}_{n_k \text{ times}}\}.$$

Then, the clearing time $\mathcal{W}_i^{\mathrm{CP}}$ is given by

$$\begin{aligned} y_1 &= c_1 \to \bar a_1,\\ y_k &= (y_{k-1} \oplus c_k) \to \bar a_k \quad \forall\, k \in \{2, \dots, i-1\},\\ \mathcal{W}_i^{\mathrm{CP}} &= (y_{i-1} \oplus c_i)^{n_i}. \end{aligned}$$

Note that $\bar a_k$ is essentially the augmented sequence of k-customer arrival times including the n_k k-customers initially waiting at time 0.

Given two sequences $y = \{y^m\}_{m=1}^{\bar m}$ and $\tilde y = \{\tilde y^m\}_{m=1}^{\tilde m}$ of not necessarily identical length, we define the relationship

$$\begin{aligned} y \le \tilde y \quad &\text{if and only if } \tilde m \le \bar m \quad \text{and}\\ y^m \le \tilde y^m \quad &\forall\, m \in \{1, \dots, \tilde m\}. \end{aligned}$$

The above relationship can be interpreted as an element-wise comparison of the two sequences, where elements equal to $+\infty$ are appended at the end of the shorter sequence so as to equalize the sequence lengths. Note in particular that if $\tilde m = 0$, then $y \le \tilde y$ for all y.

The remainder of the proof is based on the following structural properties of our operators.

Proposition 9. *Given the ordered sequences c, $\tilde c$, a, $\tilde a$, and y, the following statements hold true*:

(i) *If $\tilde c \ge c$, then $y \oplus \tilde c \ge y \oplus c$.*
(ii) *If $\tilde c \ge c$, then $\tilde c \to a \ge c \to a$.*
(iii) *If $\tilde a \le a$, then $c \to \tilde a \ge c \to a$.*

We are now ready to show that $\mathcal{W}_i^{\mathrm{CP}}$ is increasing in the service times x and decreasing in the arrival times a. Let $\mathcal{W}_i^{\mathrm{CP}\prime}$ denote the clearing time under service and arrival times given by x' and a', respectively. Then, from Proposition 8, $\mathcal{W}_i^{\mathrm{CP}\prime}$ is expressible analytically via

$$\begin{aligned} y'_1 &= c'_1 \to \bar a'_1,\\ y'_k &= (y'_{k-1} \oplus c'_k) \to \bar a'_k \quad \forall\, k \in \{2, \dots, i-1\},\\ \mathcal{W}_i^{\mathrm{CP}\prime} &= (y'_{i-1} \oplus c'_i)^{n_i}. \end{aligned}$$

Let x and a be such that $x \ge x'$ and $a \le a'$. Then, $c \ge c'$, and it follows from Proposition 9(i) that $\bar a_k \le \bar a'_k$ for all $k \in \{1, \dots, i-1\}$. Propositions 9(ii) and (iii) then imply that

$$y_1 = c_1 \to \bar a_1 \ge c'_1 \to \bar a_1 \ge c'_1 \to \bar a'_1 \ge y'_1.$$

Applying Proposition 9(i) twice yields

$$y_1 \oplus c_2 \geq y_1' \oplus c_2 \geq y_1' \oplus c_2'.$$

Fix $k \in \{2, \ldots, i-1\}$. Suppose that $y_{k-1} \oplus c_k \geq y'_{k-1} \oplus c'_k$. Then, Propositions 9(ii) and (iii) imply that $y_k \geq y'_k$. Thus, $y_k \geq y'_k$ for all $k \in \{1, \ldots, i-1\}$. Proposition 9(i) yields that $y_{i-1} \oplus c_i \geq y'_{i-1} \oplus c'_i$, and therefore $\mathcal{W}_i^{\mathrm{CP}} \geq \mathcal{W}_i^{\mathrm{CP}\prime}$, which concludes the proof. □

Proof of Theorem 4. The proof is similar to Theorem 2 and is omitted for brevity. □

Proof of Theorem 5. Fix any $i = 1, \ldots, K$. Consider a hierarchical MCMS system that comprises the first i classes and servers, but operates under FCFS. That is, an HMCMS with i classes, server parameters μ_j, $\Gamma_j^{\mathbb{X}}$, and α_j, $j = 1, \ldots, i$, and population uncertainty set

$$\mathbb{P}' = \{n \in \mathbb{R}^i \colon (n, \tilde{n}) \in \mathbb{P} \text{ for some } \tilde{n} \in \mathbb{R}^{K-i}\},$$

where servers follow FCFS. Under no arrivals, problem (8) reduces to problem (4), since we have $q = u = 0$. Therefore, we have that $\hat{W}_i^{\mathrm{CP}} = \hat{W}_i$, for all $i = 1, \ldots, K$.

We next argue that for a closed HMCMS system, the worst-case clearing time of the last class is equal for both CP and FCFS priorities.

Proposition 10. *For a hierarchical MCMS service system under no arrivals, $W_K = W_K^{\mathrm{CP}}$.*

Fix again an $i = 1, \ldots, K$ and consider a hierarchical MCMS system that comprises the first i classes and servers, but operates under FCFS, as before. Since we deal with closed systems, Proposition 10 yields that $W_i^{\mathrm{CP}} = W_i$. By Theorem 3, we obtain that $W_i \leq \hat{W}_i \leq W_i + 2\chi$. Replacing for W_i and $\hat{W}_i$, we obtain that

$$W_i^{\mathrm{CP}} \leq \hat{W}_i \leq W_i^{\mathrm{CP}} + 2\chi, \quad i = 1, \ldots, K. \quad \square$$

Proof of Lemma 3. The proof is similar to Lemma 2 and is omitted for brevity. □

Proof of Theorem 6. The proof is similar to Theorem 2 and is omitted for brevity. □

F.1. Proofs of Auxiliary Results

Proof of Proposition 3. The proof proceeds in three steps.

Step 1: *Problem* (1) *is feasible and has a finite optimal value.* The sets $\mathbb{P} \cap \mathbb{N}^K$, $\Sigma(n)$, and $\mathbb{X}_j$, $j = 1, \ldots, M$, are all nonempty by construction. It follows that problem (1) is feasible. Boundedness of the optimal value of (1) follows from boundedness of its feasible region.

Step 2: *The optimal values of problems* (1) *and* (F.1) *are equal.* First, let (n, σ, x) be feasible in (1). We construct w, q, and c such that $w_i = \mathcal{W}_i(n_1, \ldots, n_K, \sigma, x_1, \ldots, x_M)$ and (w, n, q, c) is feasible in (F.1). For all $j \in \{1, \ldots, M\}$ and $\ell \in \{1, \ldots, \bar{\ell}_j\}$, define $c_j^\ell := \sum_{k=1}^{\ell} x_j^k$. We assume without loss of generality that the elements of c are all distinct from one another and positive. All of our arguments remain valid if this assumption is relaxed at the cost of complicating notation. As in Section 2, let $\mathbb{L}_k \colon \mathbb{R}_+ \to 2^{\{1, \ldots, K\}}$, $k \in \{1, \ldots, n\}$ be multivalued functions that map time to the set of k-customers still waiting to be served. For all $j \in \{1, \ldots, M\}$ and $\ell \in \{1, \ldots, \bar{\ell}_j\}$, define

$$q_j^\ell := \begin{cases} k & \text{if } \bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell) \neq \emptyset \text{ and} \\ & \quad \arg\min\left\{\sigma(\nu) \colon \nu \in \bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell)\right\} \in \mathbb{L}_k(c_j^\ell), \\ K+1 & \text{else.} \end{cases}$$

Note that since all of the elements of σ are distinct, the minimization problem in this definition presents a unique minimizer. Also, for all $k \in \{1, \ldots, K\}$, let

$$w_k := \mathcal{W}_k(n_1, \ldots, n_K, \sigma, x_1, \ldots, x_M) = \inf\{t \geq 0 \colon |\mathbb{L}_k(t)| = 0\}.$$

Constraints (F.1b), (F.1f), and (F.1g) are clearly satisfied. Fix $j \in \{1, \ldots, M\}$ and $\ell \in \{1, \ldots, \bar{\ell}_j\}$. It follows from the definitions of w and q that if $c_j^\ell < w_{k'}$ for some $k' \in \mathbb{Q}(j)$, then $|\mathbb{L}_{k'}(c_j^\ell)| > 0$ and therefore $\bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell) \neq \emptyset$, implying that $q_j^\ell \in \mathbb{Q}(j)$. Since the choice of j and ℓ was arbitrary, constraint (F.1c) is satisfied. Fix $k \in \{1, \ldots, K\}$. Until time w_k (note that $w_k < \infty$), the total number of customers served from queue k is equal to n_k, and constraint (F.1d) is satisfied. By construction, the function $|\mathbb{L}_k(t)|$ is nonincreasing, left-continuous, with discontinuities at all instants $t \in \{t \geq 0 \colon t = c_j^\ell \text{ and } q_j^\ell = k\}$. Thus,

$$w_k = \begin{cases} \max\{c_j^\ell \colon q_j^\ell = k, j \in \mathbb{S}(k), \ell = 1, \ldots, \bar{\ell}_j\} & \text{if } |\mathbb{L}_k(0)| > 0, \\ 0 & \text{else,} \end{cases}$$

and constraint (F.1e) is satisfied. We have thus constructed a solution (w, n, q, c) feasible in (F.1) and such that $w_i = \mathcal{W}_i(n_1, \ldots, n_K, \sigma, x_1, \ldots, x_M)$.

Second, let (w, n, q, c) be feasible in (F.1). Note that the existence of such a solution is guaranteed since problem (1) is feasible (see Step 1) and we have just shown that any feasible solution to (1) can be used to construct a feasible solution to (F.1). We will construct a solution σ and x such that (n, σ, x) is feasible in (1) and $\mathcal{W}_i(n_1, \ldots, n_K, \sigma, x_1, \ldots, x_M) = w_i$. We again assume without loss of generality that the elements of c are all distinct from one another and positive. For all $j \in \{1, \ldots, M\}$ and $\ell \in \{1, \ldots, \bar{\ell}_j\}$, define $x_j^\ell := c_j^\ell - c_j^{\ell-1}$, where we use the convention that $c_j^0 = 0$. Also, define $\lambda \colon \mathbb{R}_+ \to \{1, \ldots, \sum_{k=1}^K n_k\}$ and $\lambda_k \colon \mathbb{R}_+ \to \{1, \ldots, n_k\}$, $k \in \{1, \ldots, K\}$ through

$$\lambda(t) := \sum_{\substack{j=1,\ldots,M \\ \ell=1,\ldots,\bar{\ell}_j}} \mathcal{I}(c_j^\ell \leq t \text{ and } q_j^\ell \in \mathbb{Q}(j)) \quad \text{and}$$

$$\lambda_k(t) := \sum_{\substack{j \in \mathbb{S}(k) \\ \ell=1,\ldots,\bar{\ell}_j}} \mathcal{I}(c_j^\ell \leq t \text{ and } q_j^\ell = k),$$

which count the number of all customers served by time t or the number of k-customers, respectively. Thus, if $q_j^\ell = k \in \mathbb{Q}(j)$, then the k-customer who receives the ℓth service of the jth server is the $\lambda(c_j^\ell)$th customer to be served in the system. For each $k \in \{1, \ldots, K\}$, $m \in \{1, \ldots, n_k\}$, define

$$\sigma\left(\sum_{k'=1}^{k-1} n_{k'} + m\right) := \left\{\lambda(c_j^\ell) \colon q_j^\ell = k \text{ and } \lambda_k(c_j^\ell) = m, \right. \\ \left. j \in \mathbb{S}(k), \ell \in \{1, \ldots, \bar{\ell}_j\}\right\}.$$

Thus, if $\nu = \sum_{k'=1}^{k-1} n_{k'} + m$, customer ν is the mth k-customer waiting at $t = 0$, and $\sigma(\nu)$ is the order in which he is served. By our assumption that the elements of c are all distinct from one another, σ defines a permutation of n—i.e., $\sigma \in \Sigma(n)$, and thus the second constraint in (1) is satisfied. By construction, x also satisfies the last constraint in (1). Therefore, (n, σ, x) is feasible in (1). Note in particular that under this solution, customers are sorted according to the order in which they received service under q. We now show that, $\mathcal{W}_i(n_1, \dots, n_K, \sigma, x_1, \dots, x_M) = w_i$. It suffices to show that for all $j \in \{1, \dots, M\}$, $k \in \mathbb{Q}(j)$, and $\ell \in \{1, \dots, \bar{\ell}_j\}$, it holds that

$$q_j^\ell = k \;\; \Leftrightarrow \;\; \bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell) \neq \emptyset \quad \text{and} \\ \arg\min\left\{\sigma(\nu)\colon \nu \in \bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell)\right\} \in \mathbb{L}_k(c_j^\ell), \tag{F.5}$$

so that the same customers are served under the allocation q and the permutation σ each time a server becomes available. We prove this statement by induction on the ordered sequence of server completion times c.

Fix $j' \in \{1, \dots, M\}$, and $\ell' \in \{1, \dots, \bar{\ell}_{j'}\}$. Suppose that (F.5) is true for all j and ℓ such that $c_j^\ell < c_{j'}^{\ell'}$. We first show that it must also be true for $j = j'$ and $\ell = \ell'$. It follows from (F.5) that for all $k \in \{1, \dots, K\}$ and $t \le c_{j'}^{\ell'}$, it holds that

$$\mathbb{L}_k(t) = \mathbb{L}_k(0) - \left\{\sum_{k'=1}^{k-1} n_{k'} + m\colon \\ m \in \left\{1, \dots, \sum_{\substack{j \in \mathbb{S}(k) \\ \ell = 1, \dots, \bar{\ell}_j}} \mathcal{I}(c_j^\ell < t \text{ and } q_j^\ell = k)\right\}\right\},$$

and thus

$$|\mathbb{L}_k(t)| = n_k - \sum_{\substack{j \in \mathbb{S}(k) \\ \ell = 1, \dots, \bar{\ell}_j}} \mathcal{I}(c_j^\ell < t \text{ and } q_j^\ell = k). \tag{F.6}$$

If $q_{j'}^{\ell'} = k \in \mathbb{Q}(j)$, it follows from (F.6) and from the feasibility of q in (F.1) that $|\mathbb{L}_k(c_{j'}^{\ell'})| > 0$, and therefore $\bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell) \neq \emptyset$. Moreover, it follows from the definition of λ that the first customer waiting at queue k at time $c_{j'}^{\ell'}$ under q is the $\lambda(c_{j'}^{\ell'})$th customer being served in the system. Finally, the definition of σ implies that at time $c_{j'}^{\ell'}$, all customers ν with $\sigma(\nu) < \lambda(c_{j'}^{\ell'})$ have already left the system. Thus, $\arg\min\{\sigma(\nu)\colon \nu \in \bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_{j'}^{\ell'})\} \in \mathbb{L}_k(c_{j'}^{\ell'})$ holds. If instead $q_{j'}^{\ell'} = K + 1$, then constraints (F.1c) and (F.1e) imply that $c_{j'}^{\ell'} > w_{k'}$ for all $k' \in \mathbb{Q}(j')$, and it follows from (F.6) that $|\mathbb{L}_{k'}(c_{j'}^{\ell'})| = 0$ for all $k' \in \mathbb{Q}(j)$—i.e., $\bigcup_{k' \in \mathbb{Q}(j)} \mathbb{L}_{k'}(c_j^\ell) = \emptyset$, and the right hand-side in (F.5) cannot hold. We conclude that (F.5) is true for $j = j'$ and $\ell = \ell'$.

To complete the induction, we show that (F.5) is true for the first completion time—i.e., for $j' \in \{1, \dots, M\}$ such that $c_{j'}^1 \le c_j^\ell$ for all $j \in \{1, \dots, M\}$, $\ell \in \{1, \dots, \bar{\ell}_j\}$. If $q_{j'}^1 = k \in \mathbb{Q}(j)$, then $\sigma(1 + \sum_{k'=1}^{k-1} n_{k'}) = 1$—i.e., the highest priority is a k-customer. Moreover, since by time $c_{j'}^1$, no other customer has been assigned to a server yet under q, it holds that $|\mathbb{L}_k(c_{j'}^1)| > 0$, and the right hand-side of (F.5) holds true for $j = j'$ and $\ell = 1$. If instead $q_{j'}^1 = K + 1$, constraints (F.1c) and (F.1e) combined with the fact that $c_{j'}^1 \le c_j^\ell \; \forall\, j, \ell$ imply that $n_{k'} = 0$ for all $k' \in \mathbb{Q}(j)$, a contradiction.

We conclude that (F.5) is true for all j, ℓ, and k, and therefore the completion time of queue i under the allocation q and the permutation σ are equal—i.e., $\mathcal{W}_i(n_1, \dots, n_K, \sigma, x_1, \dots, x_M) = w_i$. Consequently, problems (1) and (F.1) have the same optimal value, which is finite. □

Proof of Proposition 4. Let (w, n, y, f, c) be an optimal solution in (2) whose existence follows from the Weierstrass Theorem. Without loss of generality, we assume that constraint (2c) is active at this optimal solution. Otherwise, such an optimal solution can be readily constructed in an iterative fashion, starting from (w, n, y, f, c). We construct a feasible solution in (F.1) as follows. Let

$$(\ell', j') \in \left\{(\ell, j) \in \mathbb{R}^2\colon \sum_{k' \in \mathbb{Q}(j)} y_{k'j}^\ell = 0 \text{ and } c_j^\ell = w_i, \\ j \in \mathbb{S}(i), \ell \in \{1, \dots, \bar{\ell}_j\}\right\}.$$

Note that by the definition of $\bar{\zeta}$, the set above is never empty (otherwise it would contradict the optimality of (w, n, y, f, c) in (2)) and therefore the pair (ℓ', j') is well defined. For $j \in \{1, \dots, M\}$, $k \in \mathbb{Q}(j)$, $\ell \in \{1, \dots, \bar{\ell}_j\}$, define

$$q_j^\ell := \begin{cases} \displaystyle\sum_{k \in \mathbb{Q}(j)} k y_{kj}^\ell & \text{if } \displaystyle\sum_{k \in \mathbb{Q}(j)} y_{kj}^\ell = 1, \\ i & \text{if } j = j' \text{ and } \ell = \ell', \\ K + 1 & \text{else,} \end{cases}$$

and $\tilde{w}_k := \max\{c_j^\ell\colon q_j^\ell = k, j \in \mathbb{S}(k), \ell \in \{0, \dots, \bar{\ell}_j\}\}$. Also let $\tilde{n} := n + \mathrm{e}_i$. By definition of q and $\tilde{w}$, (F.1b) and (F.1e) are both trivially satisfied. It follows from $n + \mathrm{e}_i \in \mathbb{P} \cap \mathbb{N}^K$ that $\tilde{n} \in \mathbb{P} \cap \mathbb{N}^K$. Thus, $(\tilde{w}, n, q, c)$ satisfies constraints (F.1f) and (F.1g). In addition,

$$\sum_{\substack{\ell = 1, \dots, \bar{\ell}_j \\ j \in \mathbb{S}(i)}} \mathcal{I}(q_j^\ell = i) = 1 + \sum_{\substack{\ell = 1, \dots, \bar{\ell}_j \\ j \in \mathbb{S}(i)}} y_{ij}^\ell = n_i + 1 = \tilde{n}_i,$$

where the first and second equalities follow from the definition of q and the feasibility of y in (2), respectively. For $k \in \{1, \dots, K\}$, $k \neq i$ it holds that

$$\sum_{\substack{\ell = 1, \dots, \bar{\ell}_j \\ j \in \mathbb{S}(k)}} \mathcal{I}(q_j^\ell = k) = \sum_{\substack{\ell = 1, \dots, \bar{\ell}_j \\ j \in \mathbb{S}(k)}} y_{kj}^\ell = n_k = \tilde{n}_k.$$

Thus, (F.1d) is satisfied for all $k \in \{1, \dots, K\}$. Fix $k \in \{1, \dots, K\}$. If $k \neq i$, it directly follows from the definition of $\tilde{w}$ that $\tilde{w}_k \le w_k$. Moreover, it follows from the choice of (ℓ', j') that $\tilde{w}_i = w_i$. Thus, $\tilde{w}_k \le w_k$ for all $k \in \{1, \dots, K\}$. Fix $j \in \{1, \dots, M\}$ and $\ell \in \{1, \dots, \bar{\ell}_j\}$ and suppose that $c_j^\ell < \tilde{w}_k$ for some $k \in \mathbb{Q}(j)$. Then, $c_j^\ell < w_k$ and (2e) implies that $f_{kj}^\ell = 1$. It then follows from (2d) that $\sum_{k' \in \mathbb{Q}(j)} y_{k'j}^\ell = 1$. The definition of q then implies that $q_j^\ell \in \mathbb{Q}(j)$. Since the choice of $j \in \{1, \dots, M\}$ and $\ell \in \{1, \dots, \bar{\ell}_j\}$ was arbitrary, constraint (F.1c) is satisfied. We have thus constructed a solution $(\tilde{w}, \tilde{n}, q, c)$ feasible in (F.1) such that $\tilde{w}_i = w_i$. Thus, the optimal objective value of (F.1) is lower bounded by the optimal objective value of (2).

Suppose that there exists a solution $(\bar{w}, \bar{n}, \bar{q}, \bar{c})$ feasible in (F.1) and such that $\bar{w}_i > w_i$. Once we reach a contradiction, the proof will be complete. To this end, let

$$(\ell', j') \in \{(\ell, j) \in \mathbb{R}^2\colon \bar{q}_j^\ell = i \text{ and } \bar{c}_j^\ell = \bar{w}_i,\ j \in \mathbb{S}(i),\ \ell \in \{1, \dots, \bar{\ell}_j\}\}.$$

Note that by construction, the set above is never empty and therefore the pair (ℓ', j') is well defined. For $j \in \{1,\dots,M\}$, $k \in \mathbb{Q}(j)$, $\ell \in \{1,\dots,\bar\ell_j\}$, define

$$\bar y^{\ell}_{kj} := \begin{cases} \mathscr{I}(\bar q^{\ell}_j = i \text{ and } j \neq j' \text{ and } \ell \neq \ell') & \text{if } k = i,\\ \mathscr{I}(\bar q^{\ell}_j = k) & \text{else,}\end{cases}$$

and $\bar f^{\ell}_{kj} := \mathscr{I}(\bar c^{\ell}_j < \bar w_k)$. Also, let $\bar{\bar n} := \bar n - \mathrm{e}_i$. We now show that $(\bar w, \bar{\bar n}, \bar y, \bar f, \bar c)$ is feasible in (2). It follows from $\bar n \in \mathbb{P} \cap \mathbb{N}^K$ that $\bar{\bar n} + \mathrm{e}_i \in \mathbb{P} \cap \mathbb{N}^K$. Therefore, constraints (2g)–(2i) are satisfied. Also, it holds that

$$\sum_{k \in \mathbb{Q}(j)} \bar y^{\ell}_{kj} \leq \sum_{k \in \mathbb{Q}(j)} \mathscr{I}(\bar q^{\ell}_j = k) \leq 1,$$

where the first and second inequalities follow from the definition of $\bar y$ and the feasibility of $\bar q$ in (F.1), respectively. Thus, constraint (2b) is satisfied. Constraint (2d) is trivially satisfied if $\bar f^{\ell}_{kj} = 0$. If $\bar f^{\ell}_{kj} = 1$, then by definition it holds that $\bar c^{\ell}_j < \bar w_k$, and (F.1c) implies that $\bar q^{\ell}_j \in \mathbb{Q}(j)$—i.e., $\exists k' \in \mathbb{Q}(j)$ such that $\bar y^{\ell}_{k'j} = 1$ and constraint (2d) is satisfied in this case also. Moreover, by definition of $\bar y^{\ell}_{ij}$ and $\bar{\bar n}$, it holds that

$$\sum_{\substack{\ell=1,\dots,\bar\ell_j\\ j \in \mathbb{S}(i)}} \bar y^{\ell}_{ij} = \sum_{\substack{\ell=1,\dots,\bar\ell_j\\ j \in \mathbb{S}(i)}} \mathscr{I}(\bar q^{\ell}_j = i \text{ and } j \neq j' \text{ and } \ell \neq \ell') = \bar n_i - 1 = \bar{\bar n}_i,$$

and for $k \neq i$, it holds that

$$\sum_{\substack{\ell=1,\dots,\bar\ell_j\\ j \in \mathbb{S}(k)}} \bar y^{\ell}_{kj} = \sum_{\substack{\ell=1,\dots,\bar\ell_j\\ j \in \mathbb{S}(k)}} \mathscr{I}(\bar q^{\ell}_j = k) = \bar n_k = \bar{\bar n}_k.$$

Thus, (2c) holds true. If $\bar f^{\ell}_{kj} = 0$, then it follows from the definition of $\bar f$ that $\bar w_k \leq \bar c^{\ell}_j$, and constraint (2e) holds true. If $\bar f^{\ell}_{kj} = 1$, then constraint (2e) is trivially satisfied since $\bar\zeta$ constitutes a valid upper bound on $\bar w_k$ by construction. Finally, it follows from the definition of $\bar w_k$ that

$$\begin{aligned} \bar w_k &= \max_{\substack{\ell \in \{1,\dots,\bar\ell_j\}\\ j \in \mathbb{S}(k)}} \bar c^{\ell}_j \, \mathscr{I}(\bar q^{\ell}_j = k)\\ &\geq \max_{\substack{\ell \in \{1,\dots,\bar\ell_j\}\\ j \in \mathbb{S}(k)}} \bar c^{\ell}_j \bar y^{\ell}_{kj}\\ &\geq \max_{\substack{\ell \in \{1,\dots,\bar\ell_j\}\\ j \in \mathbb{S}(k)}} \left\{\bar c^{\ell}_j - \bar\zeta(1 - \bar y^{\ell}_{kj})\right\}\\ &\geq \bar c^{\ell}_j - \bar\zeta(1 - \bar y^{\ell}_{kj}) \quad \forall\, \ell \in \{1,\dots,\bar\ell_j\},\ j \in \{1,\dots,M\}, \end{aligned}$$

where the first equality and first inequality follow from the definitions of $\bar w_k$ and $\bar y^{\ell}_{kj}$, respectively, and where the second inequality follows from the definition of $\bar\zeta$. Therefore, constraint (2f) holds true. We have thus constructed a feasible solution $(\bar w, \bar{\bar n}, \bar y, \bar f, \bar c)$ in (2) with an objective value $\bar w_i > w_i$. This contradicts optimality of (w, n, y, f, c) in (2), and the proof is complete. □

Proof of Proposition 5. Fix $n \in \mathbb{P} \cap \mathbb{N}^K$ and consider an instance of problem (1) in which the queue population uncertainty set is given by the singleton $\{n\}$. Let W_K be the optimal value of this instance. Since the choice of $n \in \mathbb{P} \cap \mathbb{N}^K$ is arbitrary, it suffices to show that the optimal value of this instance is equal to $h_I(n)$.

Let (n, x, σ) be optimal in the new instance of problem (1) and let m_j be the number of customers served by the jth server by the clearing time W_K, for $j = 1,\dots,K$, under this solution. These numbers satisfy the following property:

$$\sum_{k=j}^{K} m_k \leq \sum_{k=j}^{K} n_k, \quad j = 1,\dots,K, \tag{F.7}$$

since the servers $j,\dots,K$, being eligible to serve customers of classes $j,\dots,K$, cannot serve more than the population of these classes.

By the clearing time definition, at W_K some server has to start serving the n_Kth K-customer; let that server be J. Consider now $\bar m \in \mathbb{R}^K$ such that

$$\bar m_j := m_j + 1, \quad j \neq J, \quad \text{and} \quad \bar m_J := m_J.$$

We will show that $(\bar m, W_K) \in \mathbb{F}_I(n)$, which will yield that $W_K \leq h_I(n)$. Clearly, $\bar m \in \mathbb{N}^K$. For $j > J$, we have that (F.7) is satisfied with strict inequality—i.e., $\sum_{k=j}^{K} m_k < \sum_{k=j}^{K} n_k$. Otherwise, the servers $j,\dots,K$, being eligible to serve customers of classes $j,\dots,K$, serve the entire population of these classes, a contradiction since server J serves one K-customer at W_K. This then implies

$$\sum_{k=j}^{K} \bar m_k = \sum_{k=j}^{K} m_k + K - j + 1 \leq \sum_{k=j}^{K} n_k + K - j.$$

For $j \leq J$, we use (F.7) to obtain

$$\sum_{k=j}^{K} m_k = \sum_{k=j}^{K} m_k + K - j \leq \sum_{k=j}^{K} n_k + K - j.$$

By Lemma 1, we can assume that service times take their worst-case values. Thus, the ℓth customer served by the j server starts receiving service at $\ell/\mu_j + \Gamma^{\mathbb{X}}_j \sqrt{\ell}$. Consequently, and by the definition of W_K and m, we get that

$$\begin{aligned} W_K &= \frac{m_J}{\mu_J} + \Gamma^{\mathbb{X}}_j \sqrt{m_J} = \frac{\bar m_J}{\mu_J} + \Gamma^{\mathbb{X}}_j \sqrt{\bar m_J};\\ W_K &\leq \frac{m_j + 1}{\mu_J} + \Gamma^{\mathbb{X}}_j \sqrt{m_j + 1} = \frac{\bar m_J}{\mu_J} + \Gamma^{\mathbb{X}}_j \sqrt{\bar m_J}, \quad j \neq J. \end{aligned}$$

To derive a contradiction and complete the proof, we assume that $h_I(n) > W_K$. Then, $\exists (\hat m, \hat w) \in \mathbb{F}_I(n)$ such that $\hat w > W_K$. Note that for all $j = 1,\dots,K$, we have that

$$\frac{m_j}{\mu_j} + \Gamma^{\mathbb{X}}_j \sqrt{m_j} \leq W_K < \hat w \leq \frac{\hat m_j}{\mu_j} + \Gamma^{\mathbb{X}}_j \sqrt{\hat m_j},$$

where the first inequality follows from the definition of m and the last by $(\hat m, \hat w) \in \mathbb{F}_I(n)$. Consequently, $m_j < \hat m_j$ for all $j = 1,\dots,K$.

Let I be the minimum index so that queues $I,\dots,K$ have cleared by W_K. Then, we can select a feasible solution (n, x, σ) that still attains the worst-case value W_K so that (F.7) is satisfied with equality for $j = I$—i.e., $\sum_{k=I}^{K} m_k = \sum_{k=I}^{K} n_k$. To see this, suppose that we have strict inequality. Since $I,\dots,K$ have cleared by W_K, then it must be that a server $r \in \{1,\dots,I-1\}$ served a customer from class $I,\dots,K$. Without loss, we can select the priority σ so that server r serves an $(I-1)$-customer instead—such a customer is guaranteed to wait, since queue

$I-1$ did not clear. By this change in assignments of customers to servers, the clearing time for queues $I,\dots,K$ can only strictly increase, leading to a contradiction of worst-case optimality of W_K, or remain the same, preserving worst-case optimality. By applying this argument recursively, we get the desired m. Then, we get a contradiction as

$$\begin{aligned}\sum_{k=I}^{K} n_k + K - I &\ge \sum_{k=I}^{K}\hat m_k \quad [\text{by } (\hat m,\hat w)\in\mathbb{F}_I(n)]\\ &\ge \sum_{k=I}^{K}(m_k+1) \quad [\text{by } \hat m_j > m_j]\\ &= \sum_{k=I}^{K} n_k + K - I + 1 \quad \left[\text{by } \sum_{k=I}^{K} m_k = \sum_{k=I}^{K} n_k\right]. \quad \square\end{aligned}$$

Proof of Proposition 6. For any $n\in\mathbb{P}$ and $(m,W)\in\mathbb{F}(n)$, let $s=\sqrt{m}$. Then, it can be readily seen that m, s, n, and W are feasible for problem (4). Thus,

$$\hat W_K \ge \max\{h(n)\colon n\in\mathbb{P}\}.$$

Conversely, for any m, s, n, and W feasible for problem (4), we have that $n\in\mathbb{P}$ and

$$W \le \frac{m_j}{\mu_j} + \Gamma_j^{\mathbb{X}} s_j \le \frac{m_j}{\mu_j} + \Gamma_j^{\mathbb{X}}\sqrt{m_j} \quad \text{and}$$
$$\sum_{k=j}^{K} m_k \le \sum_{k=j}^{K} n_k + K - j, \quad j=1,\dots,K.$$

That is, $(m,W)\in\mathbb{F}(n)$ and thus $\hat W_K \le \max\{h(n)\colon n\in\mathbb{P}\}$, completing the proof. □

Proof of Proposition 7(i). Consider any $n\in\mathbb{R}^K$.

The first inequality follows directly from the fact that $\mathbb{F}_I(n)=\mathbb{F}(n)\cap(\mathbb{N}^K\times\mathbb{R})\subset\mathbb{F}(n)$.

For the second inequality, let $(\bar m,\bar W)\in\mathbb{F}(n)$ be optimal for the maximization problem in the definition of $h(n)$—i.e., $h(n)=\bar W$. Then, it suffices to show that $(\lfloor\bar m\rfloor,\bar W-\chi)\in\mathbb{F}_I(n)$, since then by the definition of $h_I(n)$, we would have $h_I(n)\ge\bar W-\chi=h(n)-\chi$. Clearly, $\lfloor\bar m\rfloor\in\mathbb{N}$, and for any $j=1,\dots,K$, we have that

$$\sum_{k=j}^{K}\lfloor\bar m_k\rfloor \le \sum_{k=j}^{K}\bar m_k \le \sum_{k=j}^{K} n_k + K - j,$$

where the second inequality follows from $(\bar m,\bar W)\in\mathbb{F}(n)$. Finally, note that for any $j=1,\dots,K$,

$$\begin{aligned}\bar W-\chi &\le \frac{\bar m_j}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j}-\chi \quad [\text{by } (\bar m,\bar W)\in\mathbb{F}(n)]\\ &\le \frac{\lfloor\bar m_j\rfloor+1}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\lfloor\bar m_j\rfloor+1}-\chi\\ &\le \frac{\lfloor\bar m_j\rfloor+1}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\lfloor\bar m_j\rfloor}+\Gamma_j^{\mathbb{X}}-\chi \quad [\text{by } \sqrt{x+1}\le\sqrt{x}+1]\\ &\le \frac{\lfloor\bar m_j\rfloor}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\lfloor\bar m_j\rfloor} \quad \left[\text{by } \frac{1}{\mu_j}+\Gamma_j^{\mathbb{X}}\le\chi\right]. \quad \square\end{aligned}$$

Proof of Proposition 7(ii). Consider any $x,y\in\mathbb{R}^K$ with $x\le y$. Let $(\bar m,\bar W)\in\mathbb{F}_I(x)$ be optimal for the maximization problem in the definition of $h_I(x)$—i.e., $h_I(x)=\bar W$. Then, $\bar m\in\mathbb{N}^K$, and for all $j=1,\dots,K$, we have that

$$\bar W \le \frac{\bar m_j}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j} \quad \text{and}$$
$$\sum_{k=j}^{K}\bar m_k \le \sum_{k=j}^{K} x_k + K - j \le \sum_{k=j}^{K} y_k + K - j.$$

Hence, $(\bar m,\bar W)\in\mathbb{F}_I(y)$ as well, and by the definition of $h_I(y)$, we have that $h_I(y)\ge\bar W=h_I(x)$. □

Proof of Proposition 7(iii). Consider any $n\in\mathbb{R}^K$. Let $(\bar m,\bar W)\in\mathbb{F}_I(n+\mathrm{e})$ be optimal for the maximization problem in the definition of $h_I(n+\mathrm{e})$—i.e., $h_I(n+\mathrm{e})=\bar W$. We consider the following two cases.

Case 1: $\bar m_j\ge 1$ for all $j=1,\dots,K$. It suffices to show that $(\bar m-\mathrm{e},\bar W-\chi)\in\mathbb{F}_I(n)$, since then by the definition of $h_I(n)$ we would have that $h_I(n)\ge\bar W-\chi=h_I(n+\mathrm{e})-\chi$. Since $(\bar m,\bar W)\in\mathbb{F}(n+\mathrm{e})$, we get that $\bar m\in\mathbb{N}\Rightarrow(\bar m-\mathrm{e})\in\mathbb{N}$, and for any $j=1,\dots,K$, we have that

$$\sum_{k=j}^{K}(\bar m_k-1)\le\sum_{k=j}^{K}(n_k+1-1)+K-j=\sum_{k=j}^{K}n_k+K-j.$$

Note also that for any $j=1,\dots,K$,

$$\begin{aligned}\bar W-\chi &\le \frac{\bar m_j}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j}-\chi \quad [\text{by } (\bar m,\bar W)\in\mathbb{F}(n+\mathrm{e})]\\ &= \frac{\bar m_j-1}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j-1}-\chi+\frac{1}{\mu_j}+\Gamma_j^{\mathbb{X}}(\sqrt{\bar m_j}-\sqrt{\bar m_j-1})\\ &\le \frac{\bar m_j-1}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j-1}-\chi+\frac{1}{\mu_j}+\Gamma_j^{\mathbb{X}}\\ &\qquad\qquad [\text{by } \sqrt{x}-\sqrt{x-1}\le 1]\\ &\le \frac{\bar m_j-1}{\mu_j}+\Gamma_j^{\mathbb{X}}\sqrt{\bar m_j-1}. \quad \left[\text{by } \frac{1}{\mu_j}+\Gamma_j^{\mathbb{X}}\le\chi\right]\end{aligned}$$

Case 2: $\bar m_J=0$ for some $1\le J\le K$. Then, we get

$$h_I(n+\mathrm{e})=\bar W\le\frac{\bar m_J}{\mu_J}+\Gamma_J^{\mathbb{X}}\sqrt{\bar m_J}=0\le h_I(n)+\chi. \quad \square$$

Proof of Proposition 8. Recall that in a hierarchical service system under CP, a customer from any given class $k\in\{1,\dots,K\}$ will only be serviced by a server $j<k$ if the server completion time coincides with a moment when all queues 1 through $k-1$ are empty. Observe that $y_1=c_1\to\bar a_1$ corresponds to the set of times when server 1 becomes available to serve 2-customers. Fix $k\in\{2,\dots,i-1\}$. Suppose that y_{k-1} denotes the set of times when any of the servers 1 through $k-1$ becomes available to serve k-customers—i.e., the times when any server $j\in\{1,\dots,k-1\}$ completes a job and the queues j through $k-1$ are all empty. Then, the quantity $\bar c_k:=(y_{k-1}\oplus c_k)$ represents the times when any of the servers 1 through k becomes available to serve k-customers. Since these are the only servers eligible to service k-customers, the quantity $\bar c_k$ corresponds to the set of candidate k-customer service times. Accordingly, $s^t(\bar a_k,\bar c_k)$ corresponds to the number of k-customers waiting at time $\bar c_k^t$—i.e., the tth time an eligible server becomes available to service k-customers. Under a CP discipline, servers 1 through k are available to serve $(k+1)$-customers at time $\bar c_k^t$ if and only if $s^t(\bar a_k,\bar c_k)=0$. Thus,

y_k corresponds to the set of times when any of the servers 1 though k becomes available to serve $(k+1)$-customers. Therefore, the quantity $\bar{c}_i$ represents the stream of candidate i-customer service times. Since all i-customers have arrived at time 0, they will all be immediately serviced each time any of the servers 1 through i becomes available. Therefore, the n_ith i-customer will be serviced at time $\bar{c}_i^{n_i}$, which concludes the proof. □

Proof of Proposition 9(i). Let $\bar{\ell}$ and $\tilde{\ell}$ denote the lengths of c and $\tilde{c}$, respectively. Also let $\bar{m}$ denote the length of y. Since $\tilde{c} \geq c$, it follows that $\tilde{\ell} \leq \bar{\ell}$. Suppose first that $\bar{\ell} = \tilde{\ell}$. If $\tilde{\ell} = 0$, the claim follows immediately. Suppose instead that $\bar{\ell} = \tilde{\ell} > 0$. Then, $(y \oplus \tilde{c})$ and $(y \oplus c)$ have identical lengths and it suffices to perform an element by element comparison of the two sequences. Fix $\nu \in \{1, 2, \ldots, \bar{\ell} + \bar{m}\}$. Then,

$$(y \oplus \tilde{c})^\nu = \begin{cases} y^{\tilde{\lambda}} & \text{for some } \tilde{\lambda} \in \{1, \ldots, \nu\}, \text{ or} \\ \tilde{c}^{\tilde{\lambda}'} & \text{for some } \tilde{\lambda}' \in \{1, \ldots, \nu\}. \end{cases}$$

Similarly,

$$(y \oplus c)^\nu = \begin{cases} y^{\lambda} & \text{for some } \lambda \in \{1, \ldots, \nu\}, \text{ or} \\ c^{\lambda'} & \text{for some } \lambda' \in \{1, \ldots, \nu\}. \end{cases}$$

We proceed by contradiction for each possible case. Suppose that $(y \oplus \tilde{c})^\nu < (y \oplus c)^\nu$.

- If $(y \oplus \tilde{c})^\nu = y^{\tilde{\lambda}}$, then by definition of the $\oplus$ operator it follows that

$$\tilde{c}^\kappa \leq y^{\tilde{\lambda}} \quad \kappa = 1, 2, \ldots, \nu - \tilde{\lambda}, \quad \text{and} \tag{F.8a}$$
$$\tilde{c}^\kappa \geq y^{\tilde{\lambda}} \quad \kappa = \nu - \tilde{\lambda} + 1, \ldots, \bar{\ell}. \tag{F.8b}$$

—If $(y \oplus c)^\nu = y^\lambda$, then $y^{\tilde{\lambda}} < y^\lambda$ and therefore $\tilde{\lambda} < \lambda$. Moreover,

$$c^\kappa \leq y^{\lambda} \quad \kappa = 1, 2, \ldots, \nu - \lambda, \quad \text{and} \tag{F.9a}$$
$$c^\kappa \geq y^{\lambda} \quad \kappa = \nu - \lambda + 1, \ldots, \bar{\ell}. \tag{F.9b}$$

Since $\tilde{\lambda} < \lambda \leq \nu$, it follows that $\tilde{c}^{\nu-\tilde{\lambda}} \geq \tilde{c}^{\nu-\lambda}$, and thus

$$\tilde{c}^{\nu-\lambda} \leq \tilde{c}^{\nu-\tilde{\lambda}} \leq y^{\tilde{\lambda}} < y^\lambda \leq c^{\nu-\tilde{\lambda}},$$

where the second and last inequalities follow from (F.8a) with $\kappa = \nu - \tilde{\lambda}$ and (F.9b) with $\kappa = \nu - \tilde{\lambda}$, respectively. The last sequence of inequalities constitutes a contradiction.

—If $(y \oplus c)^\nu = c^{\lambda'}$, then $y^{\tilde{\lambda}} < c^{\lambda'}$. Moreover,

$$y^\kappa \leq c^{\lambda'} \quad \kappa = 1, 2, \ldots, \nu - \lambda', \quad \text{and}$$
$$y^\kappa \geq c^{\lambda'} \quad \kappa = \nu - \lambda' + 1, \ldots, \bar{m}.$$

The inequalities above imply that $\tilde{\lambda} \in \{1, 2, \ldots \nu - \lambda'\}$, so that $\tilde{\lambda} \leq \nu - \lambda'$, or equivalently $\nu - \tilde{\lambda} \geq \lambda'$. Therefore,

$$c^{\lambda'} > y^{\tilde{\lambda}} \geq \tilde{c}^{\nu-\tilde{\lambda}} \geq \tilde{c}^{\lambda'},$$

where the second inequality above follows from (F.8a) with $\kappa = \nu - \tilde{\lambda}$. The last sequence of inequalities contradicts our assumption that $\tilde{c} \geq c$.

We conclude that if $\tilde{\ell} = \bar{\ell}$ and $(y \oplus \tilde{c})^\nu = y^{\tilde{\lambda}}$, then $(y \oplus \tilde{c})^\nu \geq (y \oplus c)^\nu$.

- The proof for the case when $(y \oplus \tilde{c})^\nu = \tilde{c}^{\tilde{\lambda}'}$ mirrors exactly the case above and can thus be omitted.

If $\tilde{\ell} < \bar{\ell}$, the same proof carries through unchanged by appending $\bar{\ell} - \tilde{\ell}$ elements equal to $+\infty$ at the end of $\tilde{c}$ so as to equalize sequence lengths. We conclude that $y \oplus \tilde{c} \geq y \oplus c$. □

Before proceeding with the proof of Propositions 9(ii) and (iii), we provide a nonrecursive expression for the elements of the sequence $s(a, c)$ defined in (F.4). For any given $\tau \in \{1, \ldots, \bar{\ell}\}$, it follows from (F.4) that

$$\begin{aligned}
\sum_{t=1}^{\tau} s^t(a,c) &= \sum_{t=1}^{\tau} [s^{t-1}(a,c) - 1]^+ + z^t(a,c) - z^{t-1}(a,c) \\
&= z^\tau(a,c) + \sum_{t=0}^{\tau-1} [s^t(a,c) - 1]^+ \\
&= z^\tau(a,c) + \sum_{t=1}^{\tau-1} [s^t(a,c) - 1]^+ \quad [\text{since } s^0(a,c) = 0] \\
&= z^\tau(a,c) + \sum_{t=1}^{\tau-1} (s^t(a,c) - 1) + \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c) = 0) \\
&= z^\tau(a,c) - \tau + 1 + \sum_{t=1}^{\tau-1} s^t(a,c) + \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c) = 0),
\end{aligned}$$

which yields

$$s^\tau(a,c) = z^\tau(a,c) - \left[(\tau - 1) - \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c) = 0)\right] \quad \forall\, \tau \in \{1, \ldots, \bar{\ell}\}. \tag{F.10}$$

Equation (F.10) can be interpreted as follows: the number of people waiting to be served at time c^τ under c is equal to the difference between the total number of people that have arrived by time c^τ under a less the total number of people that have been served prior to time c^τ.

Proof of Proposition 9(ii). Let $\bar{\ell}$ and $\tilde{\ell}$ denote the lengths of c and $\tilde{c}$, respectively. Let $\tilde{q} := \tilde{c} \rightarrow a = \{\tilde{q}^m\}_{m=1}^{\tilde{m}}$ and $q := c \rightarrow a = \{q^m\}_{m=1}^{\bar{m}}$. Since $\tilde{c} \geq c$, it follows that $\tilde{\ell} \leq \bar{\ell}$. Suppose first that $\bar{\ell} = \tilde{\ell}$. If $\tilde{\ell} = 0$, then $\tilde{m} = 0$ and the claim follows immediately. Suppose $\bar{\ell} = \tilde{\ell} > 0$. We begin by showing that $\tilde{m} \leq \bar{m}$ and then demonstrate that $\tilde{q}^m \geq q^m$ for all $m \in \{1, \ldots, \tilde{m}\}$.

- If $\tilde{m} = 0$, the claim follows directly. Suppose $\tilde{m} > 0$ and let

$$\tilde{\tau} := \max\{t \in \{0, \ldots, \bar{\ell}\} \colon s^t(a, \tilde{c}) = 0\}.$$

Then, $\tilde{\tau} \geq 1$, and it follows from the definition of $\tilde{m}$ that

$$\begin{aligned}
\tilde{m} - 1 &= \sum_{t=1}^{\tilde{\tau}-1} \mathcal{I}(s^t(a,\tilde{c}) = 0) \\
&= s^{\tilde{\tau}}(a,\tilde{c}) - z^{\tilde{\tau}}(a,\tilde{c}) + \tilde{\tau} - 1 \quad [\text{from (F.10) since } \tilde{\tau} \geq 1] \\
&= \tilde{\tau} - 1 - z^{\tilde{\tau}}(a,\tilde{c}) \quad [\text{since } s^{\tilde{\tau}}(a,\tilde{c}) = 0] \\
&\leq s^{\tilde{\tau}}(a,c) + \tilde{\tau} - 1 - z^{\tilde{\tau}}(a,c) \\
&\qquad [\text{since } z^{\tilde{\tau}}(a,\tilde{c}) \geq z^{\tilde{\tau}}(a,c) \text{ and } s^{\tilde{\tau}}(a,c) \geq 0] \\
&= \sum_{t=1}^{\tilde{\tau}-1} \mathcal{I}(s^t(a,c) = 0).
\end{aligned}$$

There are two possible cases depending on the sign of $s^{\tilde{\tau}}(a,c)$. If $s^{\tilde{\tau}}(a,c) = 0$, it follows from the above that

$$\mathcal{I}(s^{\tilde{\tau}}(a,c) = 0) + \sum_{t=1}^{\tilde{\tau}-1} \mathcal{I}(s^t(a,c) = 0) \geq \tilde{m}.$$

If $s^{\tilde\tau}(a,c) > 0$, the inequality above is strict and the claim follows. In both cases, the sequence $s(a,c)$ has at least $\tilde m$ zero elements. We conclude that $\bar m \ge \tilde m$.

- We now proceed by contradiction to show that $\tilde q^m \ge q^m$ for all $m \in \{1,\dots,\tilde m\}$. Suppose that there exists $t' \in \{1,\dots,\tilde m\}$ such that $\tilde q^{t'} < q^{t'}$, while $\tilde q^t \ge q^t$ for all $t \in \{1,\dots,t'-1\}$. Then,

$$\begin{aligned} \tilde q^{t'} &= \tilde c^{\tilde\tau} \quad \text{for some } \tilde\tau \ge t', \quad \text{and} \\ q^{t'} &= c^{\tau} \quad \text{for some } \tau \ge t'. \end{aligned}$$

Therefore, $\tilde c^{\tilde\tau} < c^\tau$, from which it must hold that $\tilde\tau < \tau$. Otherwise, if $\tilde\tau \ge \tau$, then $\tilde c^{\tilde\tau} \ge \tilde c^{\tau} \ge c^\tau$, where the second inequality follows from the premise that $\tilde c \ge c$, yielding a contradiction. From the definition of $\tilde\tau$ and τ, it follows that

$$\begin{aligned} &\sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c)=0) = t'-1 \quad \text{and} \\ &\sum_{t=1}^{\tilde\tau-1} \mathcal{I}(s^t(a,\tilde c)=0) = t'-1. \end{aligned} \tag{F.11}$$

Moreover, $s^\tau(a,c) = s^{\tilde\tau}(a,\tilde c) = 0$. Then, (F.10) implies that

$$z^\tau(a,c) = \tau - t' \quad \text{and} \quad z^{\tilde\tau}(a,\tilde c) = \tilde\tau - t'. \tag{F.12}$$

Let $\tau' := \max\{t\colon c^t \le \tilde c^{\tilde\tau}\}$. Then, it must hold that $\tau' \ge \tilde\tau$. Otherwise, if $\tau' < \tilde\tau$, then from the definition of τ', $c^{\tau'} \le \tilde c^{\tilde\tau} < c^{\tilde\tau}$, which contradicts the premise that $\tilde c \ge c$. Moreover, it must hold that $\tau' < \tau$. Otherwise, if $\tau' \ge \tau$, then $\tilde c^{\tilde\tau} \ge c^{\tau'} \ge c^\tau$, a contradiction. In addition, it follows from $c^{\tau'} \le \tilde c^{\tilde\tau}$ and (F.12) that

$$z^{\tau'}(a,c) \le z^{\tilde\tau}(a,\tilde c) = \tilde\tau - t'. \tag{F.13}$$

From the nonnegativity of $s^{\tau'}(a,c)$, it follows that

$$\begin{aligned} 0 &\le s^{\tau'}(a,c) \\ &= z^{\tau'}(a,c) - (\tau'-1) + \sum_{t=1}^{\tau'-1} \mathcal{I}(s^t(a,c)=0) \quad [\text{by definition}] \\ &\le z^{\tau'}(a,c) - (\tau'-1) + \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c)=0) \quad [\tau' < \tau] \\ &\le \tilde\tau - t' - \tau' + 1 + \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c)=0) \quad [\text{from (F.13)}] \\ &= \tilde\tau - t' - \tau' + 1 + t' - 1 \quad [\text{from (\textbf{??})}] \\ &= \tilde\tau - \tau' \\ &\le 0 \quad [\text{from } \tau' \ge \tilde\tau]. \end{aligned}$$

It thus follows that the sequence of inequalities above must hold with equality. In particular, we obtain

$$\sum_{t=1}^{\tau'-1} \mathcal{I}(s^t(a,c)=0) = \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c)=0). \tag{F.14}$$

Moreover, $s^{\tau'}(a,c) = 0$, which yields

$$\sum_{t=1}^{\tau'-1} \mathcal{I}(s^t(a,c)=0) < \sum_{t=1}^{\tau'} \mathcal{I}(s^t(a,c)=0) \le \sum_{t=1}^{\tau-1} \mathcal{I}(s^t(a,c)=0)$$

and contradicts (F.14).

Since $\tilde m \le \bar m$ and $q^m \le \tilde q^m$ for all $m \in \{1,\dots,\tilde m\}$, it follows that $q \le \tilde q$, which concludes the proof for the case when $\tilde\ell = \bar\ell$. If $\tilde\ell < \bar\ell$, the same proof carries through unchanged by appending $\bar\ell - \tilde\ell$ elements equal to $+\infty$ at the end of $\tilde c$ so as to equalize sequence lengths. □

Proof of Proposition 9(iii). This proof parallels the proof of Proposition 9(ii) and is omitted. □

Proof of Proposition 10. Consider an HMCMS system under no arrivals that operates under FCFS, and let c be the servers' completion times. Let (y,n,f) be an optimal solution to problem (3). We will show that the assignments y can be taken to be compatible with class priority without loss. This will imply that $W_K \le W_K^{\text{CP}}$ for this system. Note also that since any allocation that is compatible with CP is also compatible with FCFS—under an appropriately constructed σ, as in the proof of Theorem 1—we have that $W_K \ge W_K^{\text{CP}}$. These two results will yield that $W_K = W_K^{\text{CP}}$.

To show that the assignments y can be taken to be compatible with class priority without loss, suppose that there exists an assignment implied by y that is not compatible with CP. Let τ be the largest time at which such an assignment was made. In particular, at time τ, server j became available and served a k-customer, while an i-customer was waiting, for $j \le i < k$. We denote the number of ν-customers waiting at the system at $\tau+$—i.e., immediately after the jth server started serving the k-customer—with m_ν, $\nu = 1,\dots,K$. Note that this implies that $m_i > 0$. Such an assignment is not compatible with CP indeed. All assignments at times $t > \tau$ are compatible with CP, by our choice of τ.

To show our claim, it suffices to prove that the Kth queue's clearing time would increase under the alternative (CP-compatible) assignment at time τ where server j serves an i-customer instead of a k-customer. Recall that W_K is the Kth queue's clearing time under the original assignment and let $\tilde W_K$ be the corresponding time under the alternative assignment.

Since we assumed all assignments after τ to follow CP, W_K ($\tilde W_K$) can be computed as the Kth queue's clearing time in case the system's initial queue populations were m ($m - \mathrm{e}_i + \mathrm{e}_k$) and server completion times were $d = \{c^\ell\colon c^\ell > \tau\}$ under CP. We use the notation and results derived in Lemma 2 and Proposition 8 to express W_K ($\tilde W_K$). In particular, consider the arrival processes

$$\begin{aligned} a_\nu &= \underbrace{\{0,\dots,0\}}_{m_\nu \text{ times}}, \quad \nu = 1,\dots,K; \\ \tilde a_\nu &= \underbrace{\{0,\dots,0\}}_{m_\nu - \mathcal{I}(\nu=i) + \mathcal{I}(\nu=k) \text{ times}}, \quad \nu = 1,\dots,K. \end{aligned}$$

The process a ($\tilde a$) corresponds to initial queue populations of m ($m - \mathrm{e}_i + \mathrm{e}_k$)—i.e., to the original (alternative) scenario. By Proposition 8,

$$\begin{aligned} y_1 &= d_1 \to a_1, \quad \tilde y_1 = d_1 \to \tilde a_1; \\ y_\nu &= (y_{\nu-1} \oplus d_\nu) \to a_\nu, \quad \tilde y_\nu = (\tilde y_{\nu-1} \oplus d_\nu) \to \tilde a_\nu \\ &\qquad \forall\, \nu \in \{2,\dots,K-1\}; \\ W_K &= (y_{K-1} \oplus d_K)^{m_K}, \quad \tilde W_K = (\tilde y_{K-1} \oplus d_K)^{m_K}. \end{aligned}$$

Using the monotonicity properties derived in Proposition 8, to show that $\tilde W_K \ge W_K$, it suffices to show that $\tilde y_k \ge y_k$ (in a similar fashion as in the proof of Lemma 2).

Let $y_0 := \varnothing$ and

$$h_\nu := y_{\nu-1} \oplus d_\nu, \quad \nu = 1,\dots,K.$$

Using this notation and the properties of the → operator, we get that

$$y_\nu = (y_{\nu-1} \oplus d_\nu) \longrightarrow a_\nu = h_\nu \longrightarrow a_\nu = \{h_\nu^\ell\}_{\ell \geq m_\nu}, \quad \nu = 1, \ldots, K.$$

Note that by construction of a, $\tilde{a}$, y and $\tilde{y}$, we have that

$$\tilde{y}_\nu = y_\nu, \quad \nu = 1, \ldots, i-1. \tag{F.15}$$

Combining the above, we get that

$$\begin{aligned} \tilde{y}_i &= (\tilde{y}_{i-1} \oplus d_i) \longrightarrow \tilde{a}_i \\ &= (y_{i-1} \oplus d_i) \longrightarrow \tilde{a}_i \\ &= h_i \longrightarrow \tilde{a}_i \\ &= \{h_i^\ell\}_{\ell \geq m_i - 1} \\ &= h_i^{m_i-1} \oplus y_i. \end{aligned}$$

Similarly,

$$\begin{aligned} \tilde{y}_{i+1} &= (\tilde{y}_i \oplus d_{i+1}) \longrightarrow \tilde{a}_{i+1} \\ &= (h_i^{m_i-1} \oplus y_i \oplus d_{i+1}) \longrightarrow \tilde{a}_{i+1} \\ &= (h_i^{m_i-1} \oplus h_{i+1}) \longrightarrow \tilde{a}_{i+1} \\ &= \{(h_i^{m_i-1} \oplus h_{i+1})^\ell\}_{\ell \geq m_{i+1}} \\ &= \max\{h_i^{m_i-1}, h_{i+1}^{m_{i+1}-1}\} \oplus y_{i+1}. \end{aligned}$$

Applying these operators iteratively yields that

$$\tilde{y}_{k-1} = \max_{i \leq \nu \leq k-1} \{h_\nu^{m_\nu - 1}\} \oplus y_{k-1}.$$

Finally, if we let $\eta := \max_{i \leq \nu \leq k-1}\{h_\nu^{m_\nu-1}\}$,

$$\begin{aligned} \tilde{y}_k &= (\tilde{y}_{k-1} \oplus d_k) \longrightarrow \tilde{a}_k \\ &= (\eta \oplus y_{k-1} \oplus d_k) \longrightarrow \tilde{a}_k \\ &= (\eta \oplus h_k) \longrightarrow \tilde{a}_k \\ &= \{(\eta \oplus h_k)^\ell\}_{\ell \geq m_k + 1} \\ &= \max\{\eta, h_k^{m_k}\} \oplus \{h_k^\ell\}_{\ell \geq m_k + 1} \\ &\geq \{h_k^\ell\}_{\ell \geq m_k} = y_k, \end{aligned}$$

and the proof is complete. □

Endnotes

[1] Currently, only historical estimates of wait times *aggregated* across all patients from registration to *transplant* are available. Such estimates have little utility in practice, being agnostic to patient characteristics, such as blood type and current rank in the wait list, that heavily influence actual wait time. Nor do they offer any guidance with respect to wait time until offer of a kidney of a particular quality.

[2] As we shall see, this assumption is without loss as future arrivals do not affect existing customers under FCFS.

[3] As we show in the proof of Theorem 1, formulation (2) produces the time the last customer in the ith queue leaves the system. Since we are interested in the time he receives service, we offset n_i by one.

[4] There is a well-accepted scoring system for measuring kidney quality, the kidney donor profile index, which is also used in the current national allocation policy (see Section 5).

[5] http://waittimes.alberta.ca.

[6] https://www.england.nhs.uk/statistics/2013/07/19/cancer-waiting-times-annual-report-2012-13 (July 19, 2013).

[7] To streamline exposition, we assume the worst-case completion times of all servers to be distinct. Our analysis can be readily extended otherwise, at the cost of isolating and discussing degenerate cases.

[8] Each candidate provides a list of human leukocyte antigens (HLA) that would be unacceptable in a donor in the sense that he has antibodies to such HLAs that would result in an organ rejection by his body. The probability of a candidate having unacceptable HLAs with a donor is less than 5% in the United States (http://www.ustransplant.org).

[9] When two candidates share the same HLA, they are said to be a match.

[10] Candidates are sensitized if they have unacceptable HLAs; see http://www.ustransplant.org.

[11] The most comprehensive study that leveraged all available UNOS data and experimented with a series of prediction models, including logistic regression, SVMs, boosting, CART, and Random Forests, reported error rates that varied between 21.2% and 47% (see Kim et al. 2015) in the context of liver accept/reject decisions.

[12] UNOS introduced the KDPI as standard way of measuring kidney quality in the early 2000 s, to leverage it in the KAS allocation policy; see Section 6.

[13] The Gift of Life Donor Program serves the eastern half of Pennsylvania, southern New Jersey, and Delaware.

[14] The best-quality category $j = 1$ was picked to include the narrow band of top-0% to top-6% kidneys so that all patients would be willing to accept them, as per our model specification. Indeed, all offers of kidneys in that category are accepted by available patients in our training set.

[15] There are other policy changes that we omit here since they hardly impact patient waitlist dynamics, and for the sake of brevity. For more details, see https://optn.transplant.hrsa.gov/governance/policies.

[16] UNOSNet assigns each patient an EPTS score in the range 0% to 100% that characterizes the patient's expected survivability when transplanted a median-quality kidney, as compared to other waitlisted candidates. For example, an EPTS score of 20% indicates that the patient is expected to live longer (posttransplant) than 80% of candidates.

[17] This assumption captures service perishability in kidney allocation, where unmatched kidneys are discarded, rather than preserved waiting for a matching patient to arrive. The model dynamics can be readily modified to capture cases where servers simply remain idle instead.

[18] To avoid degenerate cases, we assume that $1/\lambda_k - \Gamma_k^{\mathbb{A}} \geq 0$ for all k.

[19] In our experiments, we simulated the clearing time using the suite of applications Java Modeling Tools (JMT) (see http://jmt.sourceforge.net).

[20] These computational experiments were run on a 2.8-GHz Intel Core i7 processor machine with 24 GB of RAM, and all optimization problems were solved with CPLEX 9.1.

[21] We make the implicit assumption that 0-patients (i.e., those with top 20% EPTS score) are willing to accept kidneys only from the first server (i.e., kidneys of top quality). This is because such patients not only have priority exclusively for top-quality kidneys, but they are also in relatively better health (as reflected in their high EPTS score), affording them time to wait for top-quality kidneys. Nonetheless, relaxing this assumption is straightforward.

References

Abate J, Whitt W (1987) Transient behavior of the $M/M/1$ queue: Starting at the origin. *Queueing Systems* 2(1):41–65.

Abate J, Whitt W (1988) Transient behavior of the $M/M/1$ queue via Laplace transforms. *Adv. Appl. Probab.* 20(1):145–178.

Abate J, Whitt W (1998) Calculating transient characteristics of the Erlang loss model by numerical transform inversion. *Comm. Statist. Stochastic Models* 14(3):663–680.

Abouna GM (2008) Organ shortage crisis: Problems and possible solutions. *Transplantation Proc.* 40(1):34–38.

Akan M, Alagoz O, Ata B, Erenay FS, Said A (2012) A broader view of designing the liver allocation system. *Oper. Res.* 60(4):757–770.

Alagoz O, Maillart LM, Schaefer AJ, Roberts MS (2007) Determining the acceptance of cadaveric livers using an implicit model of the waiting list. *Oper. Res.* 55(1):24–36.

Alizadeh F, Goldfarb D (2001) Second-order cone programming. *Math. Programming* 95:3–51.

Arıkan M, Ata B, Friedewald JJ, Parker RP (2018) Enhancing kidney supply through geographic sharing in the United States. *Production Oper. Management* 27(12):2103–2121.

Aufderheide P (1999) *Communications Policy and the Public Interest: The Telecommunications Act of* 1996, The Guilford Communication Series (Guilford Press).

Bandi C, Bertsimas D (2012) Tractable stochastic analysis in high dimensions via robust optimization. *Math. Programming* 1–48.

Bandi C, Bertsimas D, Youssef N (2015) Robust queueing theory. *Oper. Res.* 63(3):676–700.

Bandi C, Bertsimas D, Youssef N (2018) Robust transient analysis of multi-server queueing systems and feed-forward networks. *Queueing Systems* 89(3-4):351–413.

Barabási A-L (2005) The origin of bursts and heavy tails in human dynamics. *Nature* 435:207–211.

Bell SL, Williams RJ (2001) Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Ann. Appl. Probab.* 11(3):608–649.

Bertsimas D, Farias VF, Trichakis N (2013) Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Oper. Res.* 61(1):73–87.

Bodur M, Luedtke JR (2017) Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Sci.* 63(7):2073–2091.

Bramson MD (2008) Stability of queueing networks. *Probab. Surveys* 5:169–345.

Choudhury GL, Whitt W (1995) Computing transient and steady-state distributions in polling models by numerical transform inversion. *IEEE Internat. Conf. Comm.*, Vol. 2, 803–809.

Choudhury GL, Lucantoni DM, Whitt W (1994) Multi-dimensional transform inversion with applications to the transient $M/G/1$ queue. *Ann. Appl. Probab.* 4(3):719–740.

Cleveland Clinic (2015) Changes to donor kidney and lung allocation programs. http://consultqd.clevelandclinic.org/2015/04/changes-to-donor-kidney-and-lung-allocation-programs.

Darling DA, Erdős P (1956) A limit theorem for the maximum of normalized sums of independent random variables. *Duke Math. J.* 23(1):143–155.

Davis AE, Mehrotra S, McElroy LM, Friedewald JJ, Skaro AI, Lapin B, Kang R, Holl JL, Abecassis MM, Ladner DP (2014) The extent and predictors of waiting time geographic disparity in kidney transplantation in the United States. *Transplantation* 97(10): 1049–1057.

Deng Y, Shen S (2016) Decomposition algorithms for optimizing multi-server appointment scheduling with chance constraints. *Math. Programming* 157(1):245–276.

Elwyn G, Edwards A, Eccles M, Rovner D (2001) Decision analysis in patient care. *Lancet* 358(9281):571–574.

Entwistle VA, Sheldon TA, Sowden A, Watt IS (1998) Evidence-informed patient choice: Practical issues of involving patients in decisions about health care technologies. *Internat. J. Tech. Assessment Health Care* 14(2):212–225.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1979, Vol. 58 (Freeman, San Francisco).

Grassmann WK (1977) Transient solutions in Markovian queueing systems. *Comput. Oper. Res.* 4(1):47–53.

Grassmann WK (1980) Transient and steady state results for two parallel queues. *Omega* 8(1):105–112.

Gross D, Shortleand JF, Thompson JM, Harris CM (2008) *Fundamentals of Queueing Theory*, 4th ed. (Wiley-Interscience, New York).

Gurvich I, Luedtke J, Tezcan T (2010) Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach. *Management Sci.* 56(7):1093–1115.

Harrison JM, López MJ (1999) Heavy traffic resource pooling in parallel-server systems. *Queueing Systems* 33(4):339–368.

Harrison JM, Van Mieghem JA (1997) Dynamic control of Brownian networks: State space collapse and equivalent workload formulations. *Ann. Appl. Probab.* 7(3):747–771.

Heyman DP, Sobel MJ (2003) *Stochastic Models in Operations Research, Volume* I: *Stochastic Processes and Operating Characteristics*, Dover Books on Computer Science Series (Dover Publications, Mineola, NY).

Horvat LD, Shariff SZ, Garg AX (2009) Global trends in the rates of living kidney donation. *Kidney Internat.* 75(10):1088–1098.

Ibrahim R, Armony M, Bassamboo A (2017) Does the past predict the future? The case of delay announcements in service systems. *Management Sci.* 63(6):1762–1780.

Jiang L, Walrand J (2010) *Scheduling and Congestion Control for Wireless and Processing Networks*, Synthesis Lectures on Communication Networks (Morgan and Claypool Publishers, Williston, VT).

Kaczynski WH, Leemis LM, Drew JH (2012) Transient queueing analysis. *INFORMS J. Comput.* 24(1):10–28.

Karlin S, McGregor J (1958) Many server queueing processes with Poisson input and exponential service times. *Pacific J. Math.* 8(1):87–118.

Keilson J (1979) *Markov Chain Models—Rarity and Exponentiality*, Applied Mathematical Sciences, Vol. 28 (Springer, New York).

Kelton WD, Law AM (1985) The transient behavior of the $M/M/s$ queue, with implications for steady-state simulation. *Oper. Res.* 33(2):378–396.

Kim S-P, Gupta D, Israni AK, Kasiske BL (2015) Accept/decline decision module for the liver simulated allocation model. *Health Care Management Sci.* 18(1):35–57.

Kong N, Schaefer AJ, Hunsaker B, Roberts MS (2010) Maximizing the efficiency of the U.S. liver allocation system through region design. *Management Sci.* 56(12):2111–2122.

Kotiah TCT (1978) Approximate transient analysis of some queuing systems. *Oper. Res.* 26(2):333–346.

Lee CP, Chertow GM, Zenios SA (2008) Optimal initiation and management of dialysis therapy. *Oper. Res.* 56(6):1428–1449.

Mamani H, Nassiri S, Wagner MR (2017) Closed-form solutions for robust inventory management. *Management Sci.* 63(5):1625–1643.

Mandelbaum A, Stolyar AL (2004) Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Oper. Res.* 52(6):836–855.

Matas AJ, Smith JM, Skeans MA, Thompson B, Gustafson SK, Stewart DE, Cherikh WS, et al. (2015) OPTN/SRTR 2013 annual data report: Kidney. *Amer. J. Transplantation* 15(Suppl. 2):1–34.

Moore SC (1975) Approximating the behavior of nonstationary single-server queues. *Oper. Res.* 23(5):1011–1032.

Odoni AR, Roth E (1983) An empirical investigation of the transient behavior of stationary queueing systems. *Oper. Res.* 31(3): 432–455.

OPTNKTC (2007) Report of the OPTN/UNOS Kidney Transplantation Committee to the Board of Directors, September 17–18.

Pinedo M (1995) *Scheduling: Theory, Algorithms, and Systems* (Prentice-Hall, Englewood Cliffs, NJ).

Plambeck EL, Ward AR (2006) Optimal control of a high-volume assemble-to-order system. *Math. Oper. Res.* 31(3):453–477.

Queyranne M, Schulz AS (1994) Polyhedral approaches to machine scheduling. Technical report, Technische Universität Berlin, Berlin.

Rider KL (1976) A simple approximation to the average queue size in the time-dependent $M/M/1$ queue. *J. ACM* 23(2):361–367.

Rothkopf MH, Oren SS (1979) A closure approximation for the nonstationary $M/M/s$ queue. *Management Sci.* 25(6):522–534.

Sandıkçi B, Maillart LM, Schaefer AJ, Alagoz O, Roberts MS (2008) Estimating the patient's price of privacy in liver transplantation. *Oper. Res.* 56(6):1393–1410.

Sandıkçi B, Maillart LM, Schaefer AJ, Roberts MS (2013) Alleviating the patient's price of privacy through a partially observable waiting list. *Management Sci.* 59(8):1836–1854.

Su X, Zenios SA (2005) Patient choice in kidney allocation: A sequential stochastic assignment model. *Oper. Res.* 53(3): 443–455.

Su X, Zenios SA (2006) Recipient choice can address the efficiency-equity trade-off in kidney transplantation: A mechanism design model. *Management Sci.* 52(11):1647–1660.

Vincent CA, Coulter A (2002) Patient safety: What about the patient? *Quality Safety Health Care* 11(1):76–80.

Whitt W, You W (2018) *Using robust queueing to expose the impact of dependence in single-server queues Oper. Res.* 66(1):184–199.

Whitt W, You W (2016) *Time-Varying Robust Queueing* (Columbia University, New York).

Xie J, Jiang Y, Xie M (2011) A temporal approach to stochastic network calculus. Working paper, Research and Innovation, Det Norske Veritas, Høvik, Norway.

Zenios SA (2005) Models for kidney allocation. Brandeau ML, Sainfort F, Pierskalla WP, eds. *Operations Research and Health Care* (Springer, Boston), 537–554.