Managing Queues with Different Resource Requirements

Noa Zychlinski[1], Carri W. Chan[2], Jing Dong[2]
[1] Faculty of Industrial Engineering and Management, Technion – Israel Institute of Technology, Haifa 32000, Israel, http://orcid.org/0000-0002-5125-3089
[2] Decision, Risk, and Operations, Columbia Business School, 3022 Broadway, New York, NY 10027
Contact: noazy@technion.ac.il (NZ), cwchan@columbia.edu (CWC), jing.dong@gsb.columbia.edu (JD)

Queueing models that are used to capture various service settings typically assume that customers require a single unit of resource (server) to be processed. However, there are many service settings where such an assumption may fail to capture the heterogeneity in resource requirements of different customers. We propose a multi-server queueing model with multiple customer classes in which customers from different classes may require different amounts of resources to be served. We study the optimal scheduling policy for such systems. To balance the holding cost, the service rate, the resource requirement, and the priority-induced idleness, we develop an index-based policy which we refer to as the idle-avoid $c\mu/m$ rule. For a two-class two-server model, where policy-induced idleness can have a big impact on system performance, we characterize cases where the idle-avoid $c\mu/m$ rule is optimal. In other cases, we establish a uniform performance bound on the amount of sub-optimality incurred by the idle-avoid $c\mu/m$ rule. For general multi-class multi-server queues, we establish the asymptotic optimality of the idle-avoid $c\mu/m$ rule in the many-server regime. For long-time horizons, we show that the idle-avoid $c\mu/m$ is throughput optimal. Our theoretical results, along with numerical experiments, provide support for the good and robust performance of the proposed policy.

Key words: Queue scheduling, different resource requirements, coupling, competitive analysis, asymptotic optimality

1. Introduction

Queueing models are widely used to model service systems. These models typically assume that customers all require a standard unit amount of the service resource (e.g., one server). However, there are many applications where customers of different types could have very different resource requirements. Motivated by such service systems, we propose a class of multi-server queueing models with multiple classes of customers where different classes of customers require different units of resources to be served. We study the optimal scheduling policy for such systems. Our analysis provides insights on how to balance holding cost, service rate, resource requirement, and idleness in such systems.

In service systems, customers from different classes may have very different service requirements. These differences may include the duration of the job, the server skill-set needed, and/or the amount of resources required. (We use the terms customers and jobs

interchangeably.) This is especially prominent in healthcare settings. For example, in the Intensive Care Unit (ICU), patients are often classified into different acuity levels, each requiring a different level of medical attention/supervision (e.g. Tarnow-Mordi et al. 2000, Masterson and Baudouin 2015). High acuity patients on ventilators may require checks every 15–30 minutes. Thus, there is usually a dedicated nurse taking care of only one such patient, during his/her shift. On the other hand, one nurse can often manage the workload required to take care of two patients at lower acuity levels. Due to its high operating costs, ICUs are often operating near or at full capacity. In this regard, ICU nurses are a very critical resource, which determines how many patients can be admitted and what level of care can be provided (Brilli et al. 2001). Although there are empirical studies showing that the workload of ICU nurses depends on the acuity level of the patients (e.g. Kim et al. 2017, O'Brien-Pallas et al. 1997, Mueller et al. 2010), to the best of our knowledge, incorporating different resource requirements based on patients' acuity has not been explicitly modeled and studied. Moreover, empirical evidence has shown that ICU workload affects the quality of care (Carayon and Gurses 2008) and work stress experienced by nurses (Fachruddin et al. 2019). Thus, carefully understanding the implications of different service requirements on admission decisions is important for both patients' safety and employee satisfaction.

Other healthcare examples include emergency services where differences in the severity of the case put different requirements on the number of medical staff (Green 1980, Sherali et al. 1991, Altay 2012); operating rooms where different types of operations have different staffing requirements; and inpatient ward units where different levels of care require different patient-to-nurse ratios (e.g. Chan et al. 2018). Different resource requirements also arise in various other service systems. For example, in customer contact centers where agents can communicate with customers via instant messaging or phone call, an agent can simultaneously handle multiple customers via messaging but only one customer via phone (Luo and Zhang 2013). Other examples include restaurants and retailing (Green 1980).

When modeling service systems as multi-class queues, the optimal scheduling policy for systems where each job requires a single server has been studied extensively in literature; see Section 1.1 for a review of related literature. The key insights derived by this body of work is the need to carefully balance the holding cost and the service rate. Our work captures an additional feature in multi-class queueing systems: different classes of jobs

have different resource requirements. Our analysis suggests that in addition to the holding cost and the service rate, we also have to take into account the resource requirements and the priority-induced idleness. How to balance these factors can be highly non-trivial. We use a combination of exact and asymptotic analyses to derive useful structural insights of the optimal scheduling policies.

Our main contributions can be summarized as follows:

• **Modeling**. We study a multi-server queuing model with multiple customer classes, where different classes require different numbers of servers to be served. This model is relevant for several service operations applications, and is especially important for the ICU setting. We allow very general demand patterns, including arbitrary time-varying arrival rates.

• **Idle-avoid $c\mu/m$ rule**. To minimize the holding cost, the general intuition is to prioritize jobs with a larger $c\mu/m$ index, where c is the holding cost, μ is the service rate, i.e., $1/\mu$ is the average service time, and m is the number of servers required. The $c\mu/m$ index can be interpreted as the cost reduction rate. Thus, maximizing it is equivalent to maximizing the instantaneous rate of reducing holding costs. However, in some cases, prioritizing jobs with a higher $c\mu/m$ index might induce idleness in the system, i.e., some servers are left idle while there are still jobs waiting in the queue. This is because the number of idle servers may not be enough to serve any of the jobs waiting in the queue. To balance the priority-induced idleness and the instantaneous cost reduction rate, we propose a modification to the classical $c\mu$ rule, which we refer to as the idle-avoid $c\mu/m$ rule. This policy can be formulated as the solution to an integer program with a penalty for idleness. We analyze this policy to provide its performance guarantee and asymptotic optimality in some settings.

— *Performance Guarantee.* In the case of a two-server two-class model, where priority-induced idleness can leave half of the capacity idle, we are able to characterize cases where the idle-avoid $c\mu/m$ rule is optimal. In general, the optimal policy can depend on the (possibly time-varying) arrival rates and the remaining time horizon. For cases where the idle-avoid $c\mu/m$ rule is not optimal, we establish that it has a competitive ratio bound of 2.

In particular, the competitive ratio analysis indicates that the performance of the idle-avoid $c\mu/m$ rule is no worse than 2 times that of the optimal policy. Note that this performance guarantee holds for arbitrary arrival rates, initial condition, and time horizon.

— *Asymptotic Optimality.* For general multi-class systems, we conduct two asymptotic modes of analysis to derive analytical insights. One is the many-server asymptotic regime, where we consider systems with increasing scales, i.e., more servers and higher arrival rates. We show that the idle-avoid $c\mu/m$ rule is asymptotically optimal in this regime. This indicates that the idle-avoid $c\mu/m$ rule performs well in large systems. Even for small systems, numerical experiments demonstrate the robustness and good performance of the idle-avoid $c\mu/m$ rule. We also study a long-run asymptotic regime, where we study the system performance as time goes to infinity under certain regularity conditions on the arrival rates. We show that the idle-avoid $c\mu/m$ rule is throughput optimal. Meanwhile, numerical experiments demonstrate that policies that do not carefully avoid idleness, e.g., the $c\mu/m$ rule, can lead to system instability.

The rest of the paper is organized as follows. We conclude this section with a brief review of the related literature. In Section 2, we introduce our model and the scheduling problem. In Section 3, we focus on a two-class two-server queue in order to understand how to balance the priority-induced idleness and the $c\mu/m$ index. In Section 4, we introduce the class of idle-aware $c\mu/m$ rules, where the idle-avoid $c\mu/m$ rule is a special case, for multi-server queues with multiple classes of customers and general resource requirements. Some asymptotic properties of the idle-avoid $c\mu/m$ rule are established in Sections 5 and 6. We present additional numerical experiments in Section 7. Lastly, we provide some concluding remarks in Section 8.

1.1. Literature Review

This paper is related to three main lines of literature. First, it is closely related to works that apply stochastic modeling to study service systems, especially healthcare systems. Second, it is related to the extensive body of works on scheduling queues with multiple classes of customers. Third, it is related to managing idleness in queues. We shall provide a brief review of the related literature along these lines.

Motivated by several service operations applications, Green (1980, 1981) is among the first to study queueing systems where different customers may require different numbers of servers. They consider a queueing system where each customer requires a random number of servers and propose a policy that prioritizes jobs with fewer server requirements. As we will see in this paper, when dealing with multiple classes of customers, a good scheduling

policy needs to carefully balance multiple factors. In addition to the resource requirement, we also need to consider the holding cost, the service rate, and the priority-induced idleness. Reiman (1991) studies the blocking probability of a multi-server loss queue where different classes of customers have different resource requirements. They assume customers are admitted into the system as long as there are enough servers available. In this work, we try to optimize the admission decision in a queue with infinite waiting room.

More generally, queueing models have been successfully applied to various healthcare applications to derive good operational policies (e.g., Yankovic and Green 2011, Armony et al. 2015). The key insight is that pertinent features of the application need to be incorporated in the model to understand the key trade-offs. Several papers study prioritization policies in various healthcare applications. For example, Mills et al. (2013) and Sun et al. (2018) focus on patient triage and prioritization under extreme resource restrictions. Saghafian et al. (2014) study complexity-augmented triage where they advocate adding a complexity-based factor to the conventional urgency-based classification in the Emergency Department (ED). Huang et al. (2015) study the optimal scheduling policy in the ED with two classes of patients: newly admitted patients and returning patients. Baron et al. (2014, 2017) study scheduling policies with strategic idleness in service networks, which are mainly motivated by healthcare systems where patients have to go through several diagnostic and treatment stations. Our work compliments this line of works by studying patient prioritization in the presence of a new feature that is very relevant to the ICU and various other service systems: different resource requirements.

How to schedule multiple classes of jobs in stochastic processing networks has been a very active area of research. For a multi-class single server queue, when the holding cost is linear, Cox and Smith (1961) is among the first to prove the optimality of a simple index-based policy, known as the $c\mu$ rule. There are various generalizations of the rule, but the optimality is mostly obtained in an asymptotic sense. For example, Van Mieghem (1995) consider general convex holding cost; Mandelbaum and Stolyar (2004) further incorporate multiple classes of servers. The key idea is to maximize the instantaneous cost reduction rate. This often leads to simple index-based policies. In contrast to the single server setting, when the network structure and resource requirements become more complicated, the management of idleness can become an important and highly non-trivial task. The first-order goal then becomes achieving system stability (Gans and van Ryzin 1997). A class of

policies known as max-weight or max-pressure policy has been established to be throughput optimal (Armony and Bambos 2003, Dai and Lin 2005). Stolyar (2004) considers the case of strongly convex holding costs and shows the max-weight policy with properly chosen parameters asymptotically minimizes the holding cost in the conventional heavy-traffic regime.

Motivated in large part by service and healthcare applications, we focus on transient cost minimization problems over an arbitrary but finite time-horizon, with arbitrary initial queue lengths and arrival patterns (e.g., time-varying arrival rates). For a two-class two-server system, we establish a uniform performance bound for an index-based policy – the idle-avoid $c\mu/m$-rule.

For small systems, policy-induced idleness play an important role on system performance. Thus, there is a more delicate trade-off between the myopic instantaneous cost reduction rate and the forward looking idleness. In terms of the analysis, one cannot rely on asymptotic arguments as much of the prior work does, we instead use constructive coupling arguments.

For more general systems, we leverage the many-server asymptotic framework to derive structural insights. When dealing with many-server systems, characterizing the optimal scheduling policy (either exactly or asymptotically) can become a lot more challenging. Harrison and Zeevi (2004) and Atar et al. (2004) study this for multi-class many-sever queues with customer abandonment in the critically loaded regime. Atar et al. (2010) derive the asymptotic optimality of a simple index-based policy, known as $c\mu/\theta$ rule, for many-server queues with abandonment in the overloaded regime. Kim et al. (2018) consider more general customer patience-time distributions beyond exponential. We refer the readers to Puha and Ward (2019) for a tutorial on scheduling policies of overloaded multi-class many-server queues with impatient customers.

Lastly, we expand a bit more on the importance of managing idleness in queues. It has long been noticed that strict priority rules can induce idleness that leads to sub-optimal performance (e.g., instability) in stochastic processing networks (Harrison 1998). The priority-induced idleness is especially prominent when having complicated resource requirements; see, for example, Rybko and Stolyar (1992), Bramson (1994). Recently, Gurvich and Van Mieghem (2017) study a network with collaboration across different types of resources and multi-tasking within those resources. There, a mismatch between the priority

level and the collaboration level can lead to inevitable capacity loss. While the dynamics and constraints in our model are different from these works, we also find that idleness can have a big impact on system performance.

2. The Model

We consider a discrete-time queueing model with N servers, I classes of customers, and an infinite buffer (queue). Time is indexed by t, $t \in \mathbb{N}$. Each Class i is characterized by the tuple $(\lambda_i, \mu_i, c_i, m_i)$. For a planning horizon of length T, $\lambda_i = (\lambda_i(t), \ldots, \lambda_i(T-1))$, where $\lambda_i(t)$ denotes the arrival probability of a Class i customer in time slot t.

Let $A_i(t)$ denote the number of Class i arrivals in period t. Then, $A_i(t) \sim$Bernoulli$(\lambda_i(t))$, independent of all other events. In each time slot, a Class i customer in service will depart with probability $\mu_i \in [0,1]$, independent of all other events. Let $D_i(t)$ denote the number of Class i departures in period t. Then, if there are v_i Class i customers in service in time slot t, $D_i(t) \sim$Binomial(v_i, μ_i). $c_i \in \mathbb{R}^+$ is the per period holding cost of a Class i customer (including during her service time). What differentiates our model from traditional queueing models is that each Class i customer requires m_i servers. In particular, if there are v_i Class i customers in service, then the total number of servers allocated to Class i is $z_i = m_i v_i$. Without loss of generality, we assume that the classes are ordered such that $m_1 \geq m_2 \geq \cdots \geq m_I$. Note that m_i's can be any positive real numbers. In practice, m_i's are in general rational numbers defined by some staff-to-customer ratio (e.g., nurse-to-patient ratio). For example $m_i = 1/3$ means that a Class i customer requires $1/3$ of a server. With a change of units, we can also define $1/3$ of a server as a unit of service capacity, in which case, $m_i = 1$. In our numerical demonstrations, we set m_i's to be integer numbers.

We focus on a discrete-time model as it facilitates our analysis of the optimal scheduling policies. Additionally, it is sufficient to capture the dynamics of a lot of healthcare systems, which is our primary motivation. For instance, admission and discharge decisions in the ICU are rarely made on a continuous basis, restricting to 15 or 30 minute intervals can reasonably capture the time scale of these decisions.

Customers within each class are served on a first-come-first-served basis. Let $X_i(t)$ denote the number of Class i customers in the system at time t and $X(t) = (X_1(t), \ldots, X_I(t))$. The scheduling policy $\pi(t) = (\pi_1(t), \ldots, \pi_I(t))$ specifies how many customers from each class

to admit in period t. We assume $\pi(t)$ is non-anticipatory. We also assume a preemptive service discipline, which imposes only the following restrictions on $\pi(t)$:

$$\sum_{i=1}^{I} m_i \pi_i(t) \leq N, \pi_i(t) \in \mathbb{N}_0 \text{ and } 0 \leq \pi_i(t) \leq X_i(t),$$

where $\mathbb{N}_0$ denotes the set of non-negative integers.

In Section 7.1, we numerically explore the impact of non-preemption.

Given these assumptions, the system under policy π evolves as:

$$X_i^{\pi}(t+1) = X_i^{\pi}(t) + A_i(t) - D_i^{\pi}(t). \tag{1}$$

In what follows, we will suppress the dependence of X and D on the scheduling policy π when it is understood from context. Note that our formulation implies that service assignments, $\pi(t)$, happen at the beginning of each period while arrivals and departures, $A_i(t)$'s and $D_i(t)$'s, happen at the end of each period.

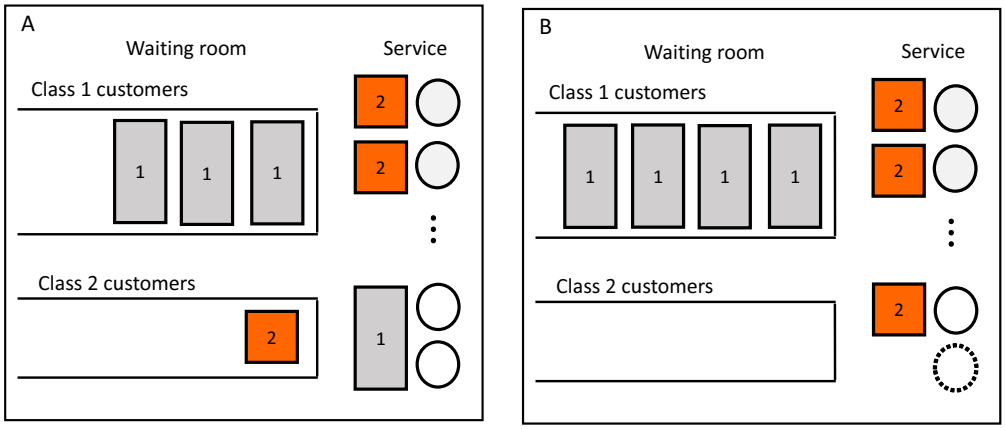
Figure 1 illustrates two possible scenarios for a system with two classes of customers and N servers. Each Class 1 customer requires two servers and each Class 2 customer requires one server (i.e., $I = 2$, $m_1 = 2$, and $m_2 = 1$).

In both scenarios illustrated in Figure 1, there are four Class 1 customers and $N-1$ Class 2 customers. In the left plot, the last two servers serve one Class 1 customer, while each of the first $N-2$ servers serves one Class 2 customer. This leaves three Class 1 customers and one Class 2 customer waiting in the queue. On the right plot, the first $N-1$ servers serve Class 2 customers. The last server is idle since there are no more Class 2 customers in the system and one server is not enough to serve a Class 1 customer. In this case, a server is idling even though there are still customers waiting in the queue.

Our objective is to find a scheduling policy that minimizes the total expected holding cost over a finite time horizon T:

$$\begin{aligned} &\min_{\pi} \sum_{t=1}^{T-1} \sum_{i=1}^{I} \mathbb{E}[c_i X_i^{\pi}(t)] + \sum_{i=1}^{I} \mathbb{E}[F_i(X_i^{\pi}(T))], \\ &\quad \text{Such that for all } t = 1, \ldots, T, \text{ and } i = 1, \ldots, I: \\ &\quad \sum_{i=1}^{I} m_i \pi_i(t) \leq N; \\ &\quad 0 \leq \pi_i(t) \leq X_i^{\pi}(t), \quad \pi_i(t) \in \mathbb{N}_0. \end{aligned} \tag{2}$$

Figure 1 Model illustration for two classes of customers.



where $\pi = (\pi(0), \pi(1), \ldots, \pi(T-1))$ and $F_i(X_i^\pi(T))$ is the terminal cost. We assume the terminal cost is proportional to the holding cost, i.e.,

$$F_i(x) = \xi c_i x, \text{ for some } \xi \in \mathbb{R}_0^+, \tag{3}$$

where $\mathbb{R}_0^+$ denotes the set of non-negative real numbers. Denote $V_0^*(x)$ as the optimal value function starting from state x at time 0 and π^* as the optimal scheduling policy.

The scheduling problem (2) is a finite-horizon Markov decision process (MDP). As the state space is countable and the action space at each state is compact, it is without loss of optimality to consider deterministic Markovian policies only (Puterman 2005). In particular, at each $t = 0, \ldots, T-1$, $\pi(t)$ can be view as a mapping from the state of Markov chain, $X(t)$, to the allocation of the servers. Note that the preemption assumption implies that, under an optimal policy, there will not be *deliberate idleness*, i.e., we would not leave m_i, $i = 1, \ldots, I$, (or more) servers idle while there are still Class i customers waiting. This does not mean there is no idleness though. As discussed earlier, there may be jobs waiting but not enough servers available for them to enter service.

We are interested in the transient scheduling problem, i.e., over a finite time-horizon with arbitrary time-varying arrival rates, in part because in healthcare applications, which is our main motivating application, time-variability in demand or random shocks like disease outbreaks or mass casualty events can lead to a demand surge and high congestion in the system. It is of interest to understand how to derive good policies in these settings.

In our subsequent analysis, an important index we will keep referring to is the $c\mu/m$ index. The $c\mu/m$ index for Class i is $c_i\mu_i/m_i$, $i = 1, 2, \ldots, I$. On average, one unit of

service capacity allocated to Class i can serve μ_i/m_i jobs over one unit of time. This reduces the holding cost by $c_i\mu_i/m_i$. Thus, the $c\mu/m$ index measures the instantaneous cost reduction rate for each class. Throughout the paper, we make the technical assumption that $(c_i\mu_i/m_i)$'s are all distinct. That is, $c_i\mu_i/m_i \neq c_j\mu_j/m_j$ for $i \neq j$. If some of these indices are equal, it is possible that there are multiple optimal scheduling policies. Non-uniqueness of the optimal policy could complicate our analysis.

3. A Two-Class Two-Server Queue

To understand the delicate balance between priority-induced idleness and instantaneous cost reduction rate, we begin by focusing on a two-class two-server model, i.e., $I = 2$ and $N = 2$. We assume that each Class 1 customer requires two servers and each Class 2 customer requires one server, i.e., $m_1 = 2$ and $m_2 = 1$. We also assume that waiting for Class 1 customers is more costly: $c_1 \geq c_2$. When considering a healthcare system, one can think of Class 1 customers (patients) as being 'sicker' than Class 2 customers (patients), thereby requiring more resources (nurses) and suffering more from waiting. In our ICU example, one unit of capacity can be viewed as 1/2 of a nurse. Thus, $m = (2, 1)$ means that a Class 1 patient requires a full nurse while a Class 2 patient requires only half a nurse time.

From the holding cost perspective, we note that if we are to maximize the instantaneous cost reduction rate, each server dedicated to serve Class 1 customers can reduce the holding cost at rate $c_1\mu_1/m_1 = c_1\mu_1/2$; each server dedicated to serve Class 2 customers can reduce the holding cost at rate $c_2\mu_2/m_2 = c_2\mu_2$. This suggests a simple strict priority rule based on the $c\mu/m$ index.

From the perspective of the processing capacity, we note that if we give strict priority to Class 2, then when $X_2(t) = 1$ and $X_1(t) \geq 1$, we can only admit one Class 2 customer into service. In this case, one server is idling while there are still Class 1 customers waiting in the queue. This can lead to substantial capacity loss if we encounter many such instances. One simple modification to avoid idleness here is to give priority to Class 1 when there is only one Class 2 customer in the system.

The above discussion motivates us to look into the following three scheduling policies: at each time epoch t, i) P_1: strict priority to Class 1, ii) P_2: strict priority to Class 2, and iii) P_2^I: a modification of P_2 that gives priority to Class 2 when $X_2(t) \geq 2$, but prioritizes

Class 1 when $X_2(t)=1$ to avoid idleness, i.e., P_2^I would prefer one Class 1 customer over one Class 2 customer.

We note that both P_1 and P_2^I avoid idleness, i.e., they are 'idle-avoid' policies, while P_2 is not. We also denote by $\mathbf{P_i}$ and $\mathbf{P_2^I}$ (in bold letters) the policies that follow P_i and P_2^I, respectively, *throughout* the time horizon, i.e., $\pi(t)=P_i(P_2^I)$ for all $t=0,\ldots,T-1$. We next study the performance of these three policies.

The analysis in this section is based on backwards induction. To facilitate the presentation, we introduce some additional notation. For $t=0,\ldots,T-2$, let

$$V_t^\pi(x)=\sum_{s=t+1}^{T-1}\sum_{i=1}^{I}\mathbb{E}[c_iX_i^\pi(s)]+\sum_{i=1}^{I}\mathbb{E}[F_i(X_i^\pi(T))]$$

denote the expected cost-to-go function in period t with $X(t)=x$ under policy π.

We define $V_{T-1}^\pi(x)=\sum_{i=1}^{I}\mathbb{E}[F_i(X_i(T))|X(T-1)=x]$. Let $S_t(x,\pi)=(S_{t,1}(x,\pi),...,S_{t,I}(x,\pi))$ denote the one step transition from state x under policy π at time t. In particular, $S_{t,i}(x,\pi)\overset{d}{=}x_i+\text{Bernoulli}(\lambda_i(t))-\text{Binomial}(\pi_i(t),\mu_i)$. We also define $C_t(x,\pi)=\sum_{i=1}^{I}c_iS_{t,i}(x,\pi)$. Then, for $t=0,\ldots,T-2$, we have

$$V_t^\pi(x)=\mathbb{E}[C_t(x,\pi)+V_{t+1}^\pi(S_t(x,\pi))].$$

3.1. Optimal Scheduling Policy

We now characterize the optimal scheduling policy for the two-class two-server system:

THEOREM 1. *For the cost minimization problem (2) with any $T>0$,*

Case 1. *When $c_2\mu_2<c_1\mu_1/2$, Policy $\mathbf{P_1}$ is optimal.*

Case 2.*: When $c_2\mu_2>c_1\mu_1/2$*

Case 2a. *When $c_1\mu_1/2<c_1\mu_1<c_2\mu_2$ and $\lambda_2(t)=0$, $\forall t$, Policy $\mathbf{P_2}$ is optimal.*

Case 2b. *When $c_1\mu_1/2<c_2\mu_2<c_1\mu_1$, Policy $\mathbf{P_2^I}$ is optimal.*

Proof. The proof of Theorem 1 is based on backwards induction and a constructive coupling argument. As the coupling arguments are similar for different cases, we only provide the analysis for Case 1 ($c_2\mu_2<c_1\mu_1/2$) here, and leave the other cases to Appendix A. We denote the policy stated in Theorem 1 by $\hat{\pi}$. In this case, $\hat{\pi}=\mathbf{P_1}$. Recall that π^* denotes the optimal policy. We shall prove that $\pi^*=\hat{\pi}$.

Base Case: $t = T-1$. We can directly derive the cost-to-go at $t = T-1$ given any server allocation, π. By definition, we have:

$$\begin{aligned} V^{\pi}_{T-1}(x) &= \mathbb{E}\left[\sum_{i=1}^{I} F_i(X_i(T)) \middle| X(T-1) = x\right] \\ &= \xi c_1 \left(x_1 + \lambda_1(T-1) - \mu_1 \pi_1(T-1)\right) + \xi c_2 \left(x_2 + \lambda_2(T-1) - \mu_2 \pi_2(T-1)\right) \end{aligned} \tag{4}$$

Due to the linearity of (4) in $\pi(T-1)$, when $c_1\mu_1 > 2c_2\mu_2$, it is optimal set $\pi_1(T-1) = x_1 \wedge 1$, $\pi_2(T-1) = x_2 \wedge (2 - 2\pi_1(T-1))$. Thus, $\pi^*_1(T-1) = P_1$.

Inductive step. Let $1 \le t \le T-1$. The inductive hypothesis is that $\pi^*(k) = \hat{\pi}(k)$ for all $k \ge t$. We will show this implies $\pi^*(t-1) = \hat{\pi}(t-1)$. The proof is by contradiction.

Suppose by contradiction, at time $t-1$, it is optimal to follow some other policy; i.e., $\pi^*(t-1) \neq \hat{\pi}(t-1)$.

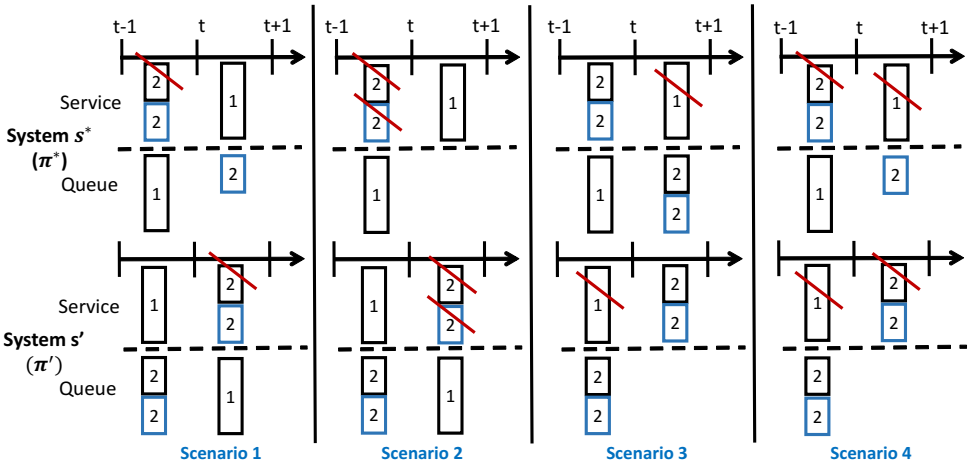
We consider two coupled systems, s^* and s', that start in the same state x at $t-1$, i.e., $X(t-1) = x$. System s^* uses policy π^* while s' uses a suboptimal policy π' that will be specified later. The coupling is induced by assuming that the two systems see exactly the same customers (the same arrival times and service time requirements path by path). We next conduct the analysis for different values of the initial state x.

- $\boldsymbol{x_1 = 0}$ **or** $\boldsymbol{x_2 = 0}$**:** π^* and $\hat{\pi}$, must coincide in this case at time $t-1$.
- $\boldsymbol{x_1 \ge 1}$ **and** $\boldsymbol{x_2 \ge 2}$**:** Because $\pi^*(t-1) \neq \hat{\pi}(t-1)$, π^* should admit two Class 2 customers at $t-1$, while $\hat{\pi}$ would admit one Class 1 customer. We construct π' such that it admits a Class 1 customer at $t-1$, and preempts this customer, if necessary, at time t to admit two Class 2 customers. From $t+1$ onward, π' will follow π^*. Considering the potential outcomes across the two systems at $t-1$, there are six scenarios:

1. Only one Class 2 customer completes service.
2. Both Class 2 customers complete service, but the one Class 1 customer does not complete.
3. Only the Class 1 customer completes service.
4. One Class 2 customer and the Class 1 customer complete service.
5. Both Class 2 customers and the Class 1 customer complete service.
6. No customer completes service.

In the s^* system, $\pi^*(t)$ will always admit the Class 1 customer, due to our inductive hypothesis. Thus, under the coupling construction, in all 6 scenarios, the two systems, s^*

Figure 2 Coupling illustration for Case 1, Scenario 1-4.



and s', are fully synchronized at $t+1$. In particular, both systems have the same customers with the same remaining service times presented (see Figure 2 for a pictorial illustration). Since the two systems follow the same policy from time $t+1$ onward, they will keep fully synchronized.

Then, the cost difference between s^* and s', $V_{t-1}^{\pi^*}(x) - V_{t-1}^{\pi'}(x)$, is the difference in the holding costs incurred at t. (We summarize the cost difference (ΔC) and the corresponding probability (Pr) for each scenario in Table 1.) Then, we have

$$V_{t-1}^{\pi^*}(x) - V_{t-1}^{\pi'}(x) = -2c_2\mu_2\prod_{i=1}^{2}(1-\mu_i) - 2c_2(1-\mu_1)\mu_2^2 + c_1\mu_1(1-\mu_2)^2$$
$$+ 2(c_1-c_2)\mu_1\mu_2(1-\mu_2) + (c_1-2c_2)\mu_1\mu_2^2 = c_1\mu_1 - 2c_2\mu_2 > 0,$$

where the last inequality follows from the condition of Case 1. This contradicts the assumption that π^* is the optimal policy.

Table 1 The cost difference and the corresponding probability for each scenario when $x_1 \geq 1$ and $x_2 \geq 2$.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
ΔC	$-c_2$	$-2c_2$	c_1	c_1-c_2	c_1-2c_2	0
Pr	$2\mu_2\prod_{i=1}^{2}(1-\mu_i)$	$(1-\mu_1)\mu_2^2$	$\mu_1(1-\mu_2)^2$	$2\mu_1\mu_2(1-\mu_2)$	$\mu_1\mu_2^2$	$(1-\mu_1)(1-\mu_2)^2$

- **$x_1 \geq 1$ and $x_2 = 1$:** To prove that admitting a Class 1 customer is preferable over admitting only one Class 2 customer, we follow the same coupling technique as in the previous case. In particular, assume by contradiction that π^* admit the Class 2 customer

at time $t-1$. We construct π' such that it admits a Class 1 customer at time $t-1$, admits the Class 2 customer at time t, and follows policy π^* from time $t+1$ onwards. In this case, there are four possible scenarios for the first time epoch $(t-1)$:

1. Only the Class 2 customer completes service.
2. Only the Class 1 customer completes service.
3. The Class 2 customer and the Class 1 customer both complete service.
4. Neither customer completes service.

Similar to before, even if there is a Class 2 arrival in $t-1$, in the s^* system $\pi^*(t)$ admits the Class 1 customer. Then, under the coupling construction, the two systems are fully synchronized at time $t+1$. Thus, the cost difference is the difference in the holding costs incurred at t, which is summarized in Table 2. Then, we have

$$V_{t-1}^{\pi^*}(x) - V_{t-1}^{\pi'}(x) = -c_2(1-\mu_1)\mu_2 + c_1\mu_1(1-\mu_2) + (c_1-c_2)\mu_1\mu_2 = c_1\mu_1 - c_2\mu_2 > 0,$$

where the last inequality follows from the condition of Case 1. This contradicts the assumption that π^* is the optimal policy.

Table 2 The cost difference and the corresponding probability for each scenario when $x_1 \geq 1$ and $x_2 = 1$.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
ΔC	$-c_2$	c_1	$c_1 - c_2$	0
Pr	$(1-\mu_1)\mu_2$	$\mu_1(1-\mu_2)$	$\mu_1\mu_2$	$(1-\mu_1)(1-\mu_2)$

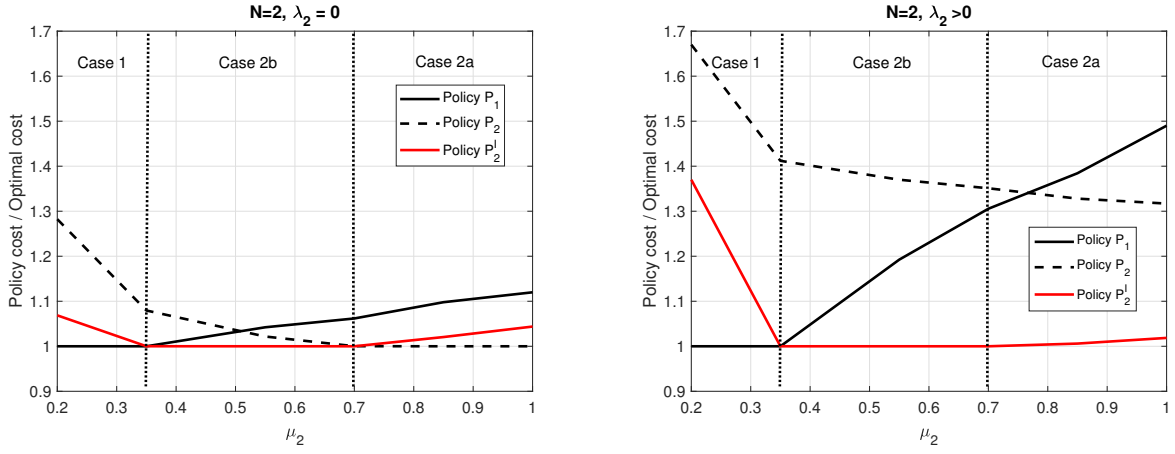
We have shown that for all possible values of state x at time $t-1$, it is optimal to follow $\hat{\pi}$. □

We next discuss the implications of Theorem 1 on how to balance idleness and instantaneous cost reduction. As $\mathbf{P_1}$ does not induce any idleness, when $c_1\mu_1/m_1 > c_2\mu_2/m_2$ (Case 1), we give strict priority to Class 1. When $c_1\mu_1/m_1 < c_2\mu_2/m_2$, we distinguish between two further cases. When $c_1\mu_1 > c_2\mu_2$ (Case 2b), from the cost perspective, admitting one Class 1 customer is preferable to admitting only one Class 2 customer. (Note that we can only admit an integer number of customers). Thus, $\mathbf{P_2^I}$ is optimal in both a processing rate sense and a cost reduction sense. When $c_1\mu_1 < c_2\mu_2$, things become more complicated. From the cost reduction perspective, even admitting only one Class 2 customer is preferable to admitting one Class 1 customer. However, admitting only one Class 2 customer

would leave one server idle in the system. In this case, we are able to show that if $\lambda_2(t) = 0$, $\mathbf{P_2}$ is optimal. When $\lambda_2(t) > 0$, it is not clear whether $\mathbf{P_2}$ is still optimal. For example, we may want to hold a single Class 2 customer in anticipation of an additional Class 2 arrival in the next period. This could help increase the processing capacity of the system and result in an overall lower cost. Whether this may be helpful will depend on a number of factors, including the length of the time horizon and the arrival probabilities in future time slots.

Figure 3 provides two numerical examples to illustrate the results of Theorem 1. The plots show the ratio between $V_0^{\pi}(2,3)$ and $V_0^{*}(2,3)$ for $\pi = \mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P_2^I}$, and for different values of μ_2. The optimal value function is calculated by solving the MDP directly and the value function for each of the three polices is estimated by simulation. We observe that in Case 2a, the optimal policy is $\mathbf{P_2}$ when $\lambda_2(t) = 0$ (left plot); however when $\lambda_2(t) > 0$ (right plot), it is not. In fact, when $\lambda_2(t) > 0$, the performance of $\mathbf{P_2^I}$ appears to be near optimal even when $c_2\mu_2 > c_1\mu_1$. Still, $\mathbf{P_2^I}$ is not exactly optimal; the optimal policy in this case is time-dependent, switching between P_2^I and P_2. Moreover, when $\lambda_2(t) = 0$ and $c_2\mu_2 > c_1\mu_1$, we observe that even though $\mathbf{P_2^I}$ leads to a higher cost than $\mathbf{P_2}$, the cost difference is fairly small. This motivates us to look more closely into $\mathbf{P_2^I}$ in the next subsection.

Figure 3 Cost ratio of each policy to the optimal policy for different values of μ_2. $T = 50$, $I = 2$, $m = (2,1)$, $\mu_1 = 0.35$, $c = (1, 0.5)$, $\xi = 5$, $N = 2$, $X(0) = (N, N)$. In the left plot $\lambda(t) = (0.33, 0)$, for $t = 1, \ldots, T-1$; In the right plot $\lambda(t) = (0.2, 0.3)$, for $t = 1, \ldots, T-1$.



3.2. A Uniform Performance Bound when $c_1\mu_1 < c_2\mu_2$ and $\lambda_2(t) \geq 0$

In this section, we analyze the performance of $\mathbf{P_2^I}$ when $c_2\mu_2 > c_1\mu_1$ (Case 2a) but with $\lambda_2(t) \geq 0$. The following theorem establishes an upper bound on the competitive ratio for $\mathbf{P_2^I}$, i.e., the ratio between the cost under $\mathbf{P_2^I}$ and the cost under the optimal policy.

THEOREM 2. *When $c_1 \geq c_2$, $c_1\mu_1 < c_2\mu_2$, for any state x, $t \in \{0, 1, \ldots, T-1\}$,*

$$\frac{V_t^{\mathbf{P_2^I}}(x)}{V_t^{\pi^*}(x)} \leq 2.$$

The proof of Theorem 2 can be found in Appendix B.2. The significance of the result in Theorem 2 is that we allow for arbitrary values for the time horizon, initial state, and arrival probabilities. It can be observed from solving the MDP that when $\lambda_2(t) \neq 0$, the optimal policy can be highly sensitive to the value of $\lambda_2(t)$'s. Moreover, even for time-homogeneous $\lambda_2(t)$'s, the optimal policy can be time-dependent. On the other hand, $\mathbf{P_2^I}$ does not depend on t or $\lambda(t)$. In addition, while the optimal policy or other benchmark policies (e.g., max-weight) may require full queue length information, $\mathbf{P_2^I}$ requires very minimal system state information, i.e., whether there are two or more Class 2 customers waiting. In healthcare applications, it can sometimes be hard to get accurate system state information. For instance, the patients waiting for ICU admission may be waiting in different wards or in other hospitals, so while it may be straightforward to know whether there are patients waiting, it may be difficult to precisely quantify the exact number of patients of each type. These desirable properties suggest that $\mathbf{P_2^I}$ is robust and easy to implement in practice.

We conclude this section with two remarks.

REMARK 1. The bound for the competitive ratio in Theorem 2 is tight, in the sense that we can find problem instances where this ratio is exactly 2. For example, at $T-1$, if $x_1(T-1) \geq 1$, $x_2(T-1) = 1$, and $\xi > 0$,

$$\frac{V_{T-1}^{\mathbf{P_2^I}}(x)}{V_{T-1}^{\mathbf{P_2}}(x)} = \frac{c_1(x_1 - \mu_1 + \lambda_1(T-1)) + c_2(1 + \lambda_2(T-1))}{c_1(x_1 + \lambda_1(T-1)) + c_2(1 - \mu_2 + \lambda_2(T-1))}. \tag{5}$$

The ratio in (5) can be made arbitrarily close to 2, if $\mu_2 = 1$, $c_1 = c_2$, $\lambda_1(T-1) = \lambda_2(T-1) = 0$, and $\mu_1 \to 0$.

On the other hand, as we will demonstrate in subsequent numerical experiments, in most problem instances, the competitive ratio is much smaller than 2.

REMARK 2. The model and ensuing analysis considered in this section applies directly to other two-class systems with $m_1/m_2 = 2$ and $N = m_1$. This can be achieved with a simple change of variables where everything is re-scaled by m_2: $\hat{X}_i = X_i/m_2$, $i = 1, 2$, and $\hat{N} = N/m_2$.

4. Idle-Aware $c\mu/m$ Rule

From our analysis in Section 3, we note that to design a good scheduling policy for queues with different resource requirements, we need to carefully balance the $c\mu/m$ index (the 'myopic' instantaneous cost reduction rate) and the priority-induced idleness. For general multi-class queues with different resource requirements, we propose a class of policies: the **idle-aware $c\mu/m$ rule**, defined as the optimal solution of an integer program (6), which maps the state of the system to an allocation of the servers to each class.

Let $x = (x_1, \ldots, x_I)$ denote the state of the system and $z = (z_1, \ldots, z_I)$ denote the number of servers allocated to each class at a time epoch t. The integer program (IP) is defined as

$$\begin{aligned} \max \ & R(z) := \sum_{i=1}^{I} \frac{c_i \mu_i}{m_i} z_i + \Gamma \sum_{i=1}^{I} z_i \\ \text{s.t.} \ & \sum_{i=1}^{I} z_i \le N \\ & 0 \le z_i \le x_i m_i, \quad i = 1, \ldots, I \\ & z_i/m_i \in \mathbb{N}_0, \quad i = 1, \ldots, I, \end{aligned} \tag{6}$$

where $\Gamma \ge 0$ is a tuning parameter that penalize the priority-induced idleness.

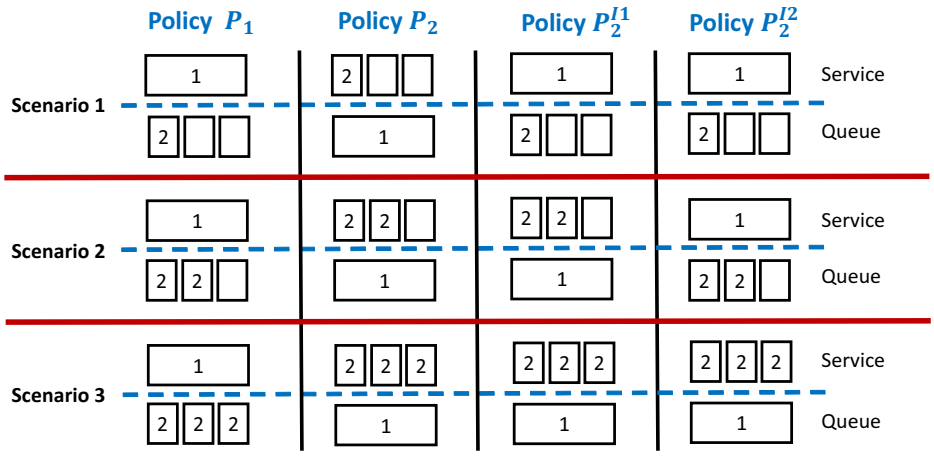
As special cases of (6), note that when $\Gamma = 0$, we prioritize according to the $c\mu/m$-index only. We refer to this special case as the **$c\mu/m$ rule**, which resembles the classical $c\mu$ rule. When Γ is large enough, i.e., $\Gamma > N \sum_{i=1}^{I} c_i\mu_i/m_i$, our first-order goal is to maximize the server utilization $\sum_{i=1}^{I} z_i$, which is equivalent to minimizing idleness, i.e., $N - \sum_{i=1}^{I} z_i$. Then, among all policies that avoid idleness, we choose the one that maximizes the $c\mu/m$ index. We refer to this special case as the **idle-avoid $c\mu/m$ rule**.

For the two-class two-server model studied in Section 3, the $c\mu/m$ rule takes the form of the policies characterized in Theorem 1. We observe from the right plot in Figure 3 that the $c\mu/m$ rule can be highly sub-optimal when $c_1\mu_1 < c_2\mu_2$ and $\lambda_2(t) \neq 0$. The idle-avoid $c\mu/m$ rule takes the following form: when $c_1\mu_1/m_1 > c_2\mu_2/m_2$, we apply P_1; when $c_1\mu_1/m_1 < c_2\mu_2/m_2$, we apply P_2^I. Combining the results in Theorems 1 and 2, we note that

regardless of the arrival rates, the initial condition, and the time horizon, when $c_1\mu_1/2 > c_2\mu_2$ or $c_1\mu_1/2 < c_2\mu_2 < c_1\mu_1$, the idle-avoid $c\mu/m$ rule is optimal; when $c_1\mu_1 < c_2\mu_2$, the idle-avoid $c\mu/m$ rule has a competitive ratio of at most 2. Thus, the idle-avoid $c\mu/m$ rule achieves good and robust performance. The analysis of this special instance of our model provides a theoretical basis for the importance of considering idleness – especially in small systems.

For more general systems, with different values of $\Gamma \in \left(0, N\sum_{i=1}^{I} c_i\mu_i/m_i\right)$, we may incur different levels of idleness. To see this, consider a two-class three-server model with $m_1 = 3$ and $m_2 = 1$. Figure 4 presents four possible scheduling policies for this model under three different scenarios of system state. Policy P_i gives strict priority to Class i, $i = 1, 2$. Policy P_2^{I1} tends to prioritize Class 2, but prefers admitting one Class 1 customer to admitting only one Class 2 customer. Policy P_2^{I2} tends to prioritize Class 2, but prefers admitting one Class 1 customer to admitting two or less Class 2 customers. Note that P_2, P_2^{I1} , and P_2^{I2} incur different levels of idleness. When solving the IP (6), if $c_1\mu_1/m_1 > c_2\mu_2/m_2$, P_1 is the optimal solution for any $\Gamma \geq 0$. However, when $c_1\mu_1 < c_2\mu_2$, the optimal solution depends on the value Γ. In particular, for small values of Γ, e.g., $\Gamma = 0$, P_2 is the optimal solution. For moderate values of Γ, P_2^{I1} is the optimal solution. For large values of Γ, P_2^{I2} is the optimal solution.

Figure 4 Possible policies for a three-server system having two classes of customers in which $m_1 = 3$ and $m_2 = 1$.



The idle-aware $c\mu/m$ rule with parameter Γ provides a lot of flexibility to determine the precise balance between the 'myopic' instantaneous cost-reduction rate and the 'forward-looking' priority-induced idleness. However, the optimal value of Γ can be quite different

for different systems. The general intuition is that when the system is very lightly loaded, we can put more weight on the $c\mu/m$ index (via a smaller Γ), while when the system is critically loaded, we should put more weight on avoiding idleness (via a larger Γ).

For systems with constant arrival rates, we define the nominal traffic intensity of the system as

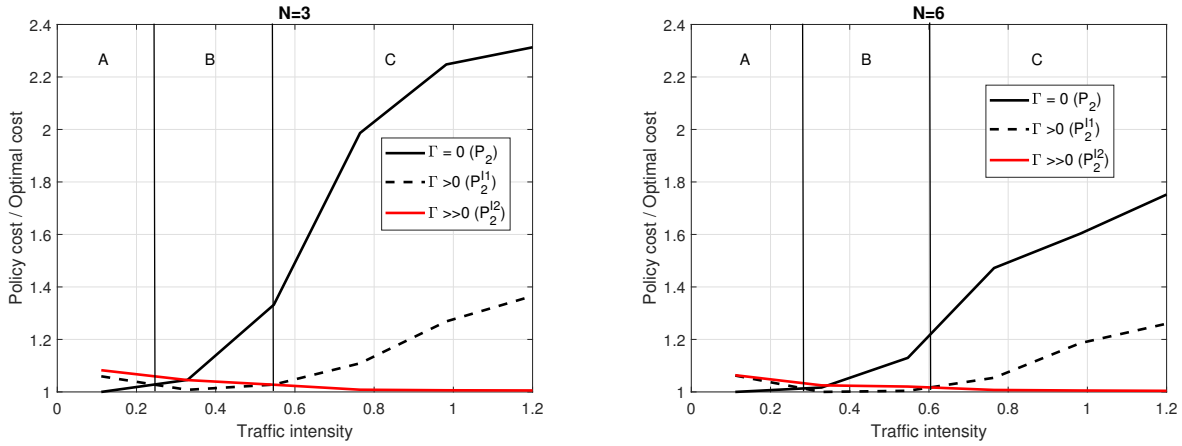
$$\rho = \sum_{i=1}^{I} \frac{\lambda_i m_i}{\mu_i N}. \tag{7}$$

Since we study finite horizon scheduling problems, we allow $\rho \geq 1$. Figure 5 plots the ratio between the cost under the idle-aware $c\mu/m$ rule with different values of Γ to the optimal cost for different traffic intensity levels. We study a two-class model with $m = (3, 1)$. In addition to $N = 3$ (left plot), we also test a larger system with $N = 6$ (right plot). To scale up the arrival rate properly with N, we assume $A_i(t) \sim Binomial(\eta, \lambda_i(t))$, where $\eta = 1$ when $N = 3$ and $\eta = 2$ when $N = 6$. We set the holding costs and the service rates such that $c_2\mu_2 > c_1\mu_1$. Three different values of Γ (idle awareness levels) are considered: $\Gamma = 0$ results in P_2, $\Gamma = 0.25$ (denoted as $\Gamma > 0$) results in P_2^{I1}, and $\Gamma = 100$ (denoted as $\Gamma \gg 0$) results in P_2^{I2}. We observe that when the traffic intensity is low, $\mathbf{P_2}$ achieves the lowest cost among the three idle-aware $c\mu/m$ policies. For moderate values of traffic intensity, $\mathbf{P_2^{I1}}$ performs the best, while for high values of traffic intensity, $\mathbf{P_2^{I2}}$ performs the best. More importantly, we note that when ρ is large (i.e., > 0.55 in the left plot and > 0.6 in the right plot), $\mathbf{P_2}$ and $\mathbf{P_2^{I1}}$ can lead to highly sub-optimal performance. On the other hand, $\mathbf{P_2^{I2}}$ achieves competitive performance across all values of traffic intensities.

For general multi-class multi-server systems, charactering the optimal scheduling policy is quite challenging. First, the coupling technique we utilized in Section 3 quickly becomes prohibitively tedious with too many scenarios to consider. Due to similar reasons, the exact optimality of the $c\mu$ rule is also restricted to the single server setting (Buyukkoc et al. 1985). Second, solving the MDP (2) exactly suffers from the curse of dimensionality (Papadimitriou and Tsitsiklis 1999).

When restricting to the class of idle-aware $c\mu/m$ rules, we observe from extensive numerical experiments that the optimal value of Γ can be highly sensitive to system parameters and using a small Γ can sometimes lead to substantial sub-optimality. On the other hand, the idle-avoid $c\mu/m$ rule in general leads to robust and near-optimal performance. Thus, we suggest using the idle-avoid $c\mu/m$ rule, i.e., setting $\Gamma > N \sum_{i=1}^{I} c_i\mu_i/m_i$, in practice.

Figure 5 Cost ratio of each policy to the optimal one for different traffic intensities where $N = 3\eta$, $\eta = 1, 2$. Here, $T = 50$, $m = (3, 1)$, the arrivals $A_i(t) \sim Binomial(\eta, \lambda_i)$, where $\lambda = k(1/24, 1/12)$ and k varies between 1 and 10.8, $\mu = (0.5, 1)$, $c = (1, 0.8)$, $\xi = 5$, $X(0) = (N/3, N-1)$. In Area A, $\mathbf{P_2}$ is the optimal optimal idle-aware $c\mu/m$ policy; in Area B, $\mathbf{P_2^{I1}}$ is the optimal idle-aware $c\mu/m$ policy, and in Area C, $\mathbf{P_2^{I2}}$ is the optimal idle-aware $c\mu/m$ policy.



In what follows, we take two asymptotic approaches to derive some theoretical insights into the performance of idle-avoid $c\mu/m$ rule. One approach focuses on the original finite-horizon planning problem with arbitrary arrival rates, but studies very large systems (Section 5). In particular, we take a many-server asymptotic mode of analysis where we scale up the arrival rates and the number of servers, while keeping the service requirements fixed. Note that when m_i's are fixed, scaling up the number of servers will lead to almost negligible policy-induced idleness. Take the two-class system with $m = (2, 1)$ as an example. When $N = 2$, strict priority to Class 2 can lead to $1/2$ of the capacity to be 'wasted'. When $N = 100$, strict priority to Class 2 can only cause $1/100$ of the capacity to be 'wasted'. Thus, our first result is a somewhat 'negative' result, showing that the class idle-aware $c\mu/m$ rules with any $\Gamma \geq 0$ is asymptotically optimal in the many-server regime. This indicates that in large systems, when $m_1 \ll N$, the policy-induced idleness plays a less important role. It also indicates that the idle-avoid $c\mu/m$ rule has near-optimal performance in these systems.

The second approach takes the large-time horizon limit, i.e., $T \to \infty$ (Section 6). We impose extra regularity conditions on the arrival probabilities and look at the stability of the system. We show that the idle-avoid $c\mu/m$ policy is throughput optimal, while the $c\mu/m$ rule and other idle-aware $c\mu/m$ rules can lead to instability. This result further

justifies our suggestion of employing the idle-avoid $c\mu/m$ rule in practice, because the other idle-aware $c\mu/m$ rules can lead to arbitrarily bad performances when planning over a long time horizon.

We conclude this section with another numerical illustration for the performance of the idle-avoid $c\mu/m$ rule. In Figure 6, we plot the cost ratio between the idle-avoid $c\mu/m$ rule and the optimal policy for systems of different sizes. As N increases, we scale up the arrival rates proportionally using an appropriate Binomial distribution. We consider both time-homogeneous (top plots) and time-varying (bottom plots) arrivals. In the the left plots, $m=(2,1)$ ($\mathbf{P_2^I}$ is the idle-avoid $c\mu/m$ rule), and in the right plots, $m=(3,1)$ ($\mathbf{P_2^{I2}}$ is the idle-avoid $c\mu/m$ rule). We randomly sample the arrival probability from $U[0,1]$. In the upper-panel plots, the arrival probabilities are drawn at time 0 and kept as constants for all $t\geq 0$. In the lower-panel plots, the arrival probabilities are updated (drawn randomly from $U[0,1]$) every 10 time slots. We report the maximal and average cost ratios among 50 randomly drawn problem instances. We observe in the left plots that when $m=(2,1)$ and $N=2$, the maximal ratio can go up to 2 as suggested by Theorem 2. However, the average ratio is much smaller, i.e., slightly larger than 1.2. Moreover, the ratios are decreasing in N. We also observe in the right plots that, when $m=(3,1)$ and $N=3$, the maximal ratio can go above 2, but the average ratio is still around 1.2. In addition, as the system size increases, both the maximal and average cost ratios are getting closer and closer to 1.

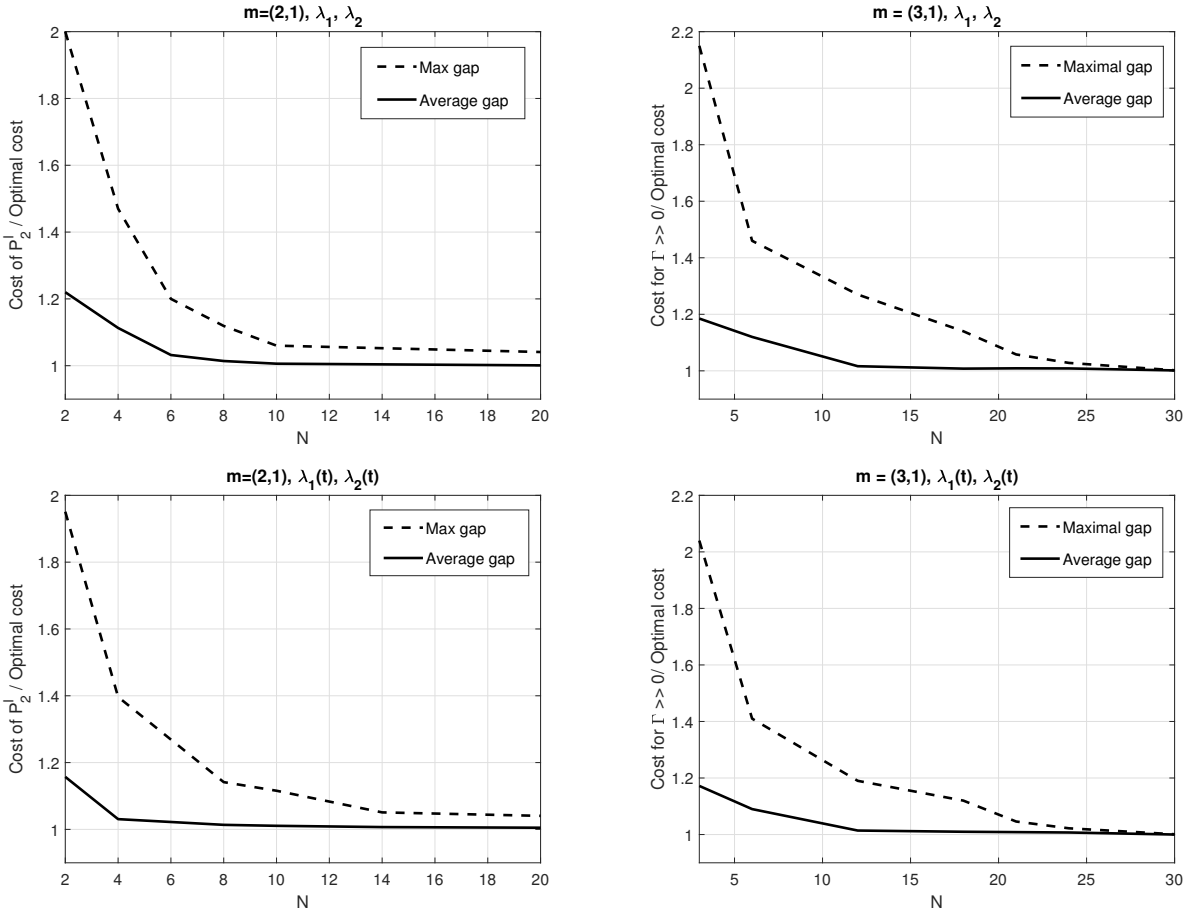
5. Asymptotic Optimality of Idle-Aware $c\mu/m$ Rule

In this section, we study the asymptotic performance of idle-aware $c\mu/m$ policies in a many-server asymptotic regime. This provides important insights into the performance of the scheduling policies in large systems with many servers and $\max_{1\leq i\leq I} m_i \ll N$.

We still focus on transient performance, i.e., over a finite time horizon and with arbitrary time-varying arrival rates.

Consider a sequence of systems indexed by η. We scale up both the number of servers and the arrival rates with η while keeping the service rates and the resource requirements fixed. In particular, for the η-th system, there are $N^\eta = N\eta$ servers and the number of Class i arrivals in the t-th epoch $A_i^\eta(t) \sim \text{Binomial}(\eta, \lambda_i(t))$. We use the superscript η to denote processes related to the η-th system. For example, $X_i^\eta(t)$ is the number of Class i customers in the η-th system at time t , $D_i^\eta(t)$ is the number of Class i departures in time

Figure 6 Optimization gap – idle avoid $c\mu/m$ rule vs. optimal policy for for different values of N: average and worst case scenario. Here, $T=50$, $I=2$, $A_i(t)\sim Binomial(\eta,\lambda_i(t))$, $\eta=1,\ldots,10$, $c=(1,1)$, and $\xi=5$. On the left, $N=2\eta$, $m=(2,1)$, $\mu=(0.01,1)$, $X(0)=(\lfloor N/2\rfloor,N-1)$; on the right, $N=3\eta$, $m=(3,1)$, $\mu=(0.09,1)$, $X(0)=(\lfloor N/3\rfloor,N-2)$. $\lambda_1(t),\lambda_2(t)\sim U[0,1]$. In the top plots, the arrival probabilities are sampled at time zero and kept as constants throughout the horizon. In the bottom plots the arrival probabilities vary every 10 time slots by drawing new samples from $U[0,1]$.



epoch t. Note that under policy π^η, $D_i^\eta(t)\sim$Binomial$(\pi_i^\eta(t),\mu_i)$. We also write R^η as the corresponding policy IP for the η-th system. For a fixed $\Gamma\geq 0$, we denote the sequence of idle-aware $c\mu/m$ rules as $(\pi^{\mathrm{IP}(\Gamma),\eta})_{\eta\geq 0}$. For a general Markovian scheduling policy π^η, it can be written as a mapping from the state of the system, x^η, to an allocation of the servers, $z^\eta=(z_1^\eta,\ldots,z_I^\eta)$. We denote $\psi^\eta:=(\psi_1^\eta,\ldots,\psi_I^\eta)$ as the corresponding mapping. In particular, $\psi^\eta:\mathbb{N}_0^I\to\mathcal{Z}$, where $\mathcal{Z}=\inf\{z\in\mathbb{N}_0^I:\sum_{i=1}^I z_i\leq N\}$, and when $X^\eta(t)=x^\eta$, $\pi_i^\eta(t)=\psi_i^\eta(x^\eta;t)/m_i$ for $i=1\ldots,I$.

We further define the fluid-scaled processes

$$\bar{X}^\eta = X^\eta/\eta, \quad \bar{A}^\eta = A^\eta/\eta \quad \bar{D}^\eta = D^\eta/\eta.$$

Let $\bar{\psi}^\eta(x^\eta/\eta; t) = \psi^\eta(x^\eta; t)/\eta$. We next define the convergence of a sequence of policies.

DEFINITION 1. We define $\bar{\psi}^\eta \to \bar{\psi}$, if for any sequence of $(\bar{x}^\eta)_{\eta\geq 1}$, $\bar{x}^\eta \in (0, 1/\eta, 2/\eta, \dots)^I$, satisfying $\bar{x}^\eta \to x$ as $\eta \to \infty$, we have $\bar{\psi}^\eta(\bar{x}^\eta) \to \bar{\psi}(x)$ as $\eta \to \infty$.

We use the fluid scaling for our analysis because the corresponding fluid limit is deterministic and provides a good approximation for the first-order mean dynamics of the system, especially for transient control problems where the demand fluctuations are $O(\eta)$.

LEMMA 1. *For a sequence of scheduling policies ψ^η, if $\bar{X}^\eta(0) \Rightarrow \bar{x}(0) \in \mathbb{R}_0^I$ and $\bar{\psi}^\eta \to \bar{\psi}$ as $\eta \to \infty$, then for any $T \geq 1$,*

$$\bar{X}^\eta \Rightarrow \bar{x} \text{ uniformly on } [0, T] \text{ as } \eta \to \infty,$$

where $\bar{x} = (\bar{x}_1, \dots, \bar{x}_I)$ is a discrete dynamical system satisfying

$$\bar{x}_i(t+1) = \bar{x}_i(t) + \lambda_i(t) - \mu_i \bar{\psi}_i(\bar{x}(t))/m_i, \quad \text{for } i = 1, \dots, I.$$

We define the fluid analogue to the MDP (2) as

$$\begin{aligned}
\min_{\bar{\pi}} \ & \bar{V}_0^{\bar{\pi}}(x) := \sum_{t=1}^{T-1} \sum_{i=1}^{I} c_i \bar{x}_i(t) + \sum_{i=1}^{I} F_i(\bar{x}_i(T)), \\
& \text{Such that for all } t = 1, \dots, T, \text{ and } i = 1, \dots, I: \\
& \bar{x}_i(t) = \bar{x}_i(t-1) + \lambda_i(t-1) - \mu_i \bar{\pi}_i(t-1) \text{ with } \bar{x}_i(0) = x_i; \\
& \sum_{i=1}^{I} m_i \bar{\pi}_i(t) \leq N; \\
& 0 \leq \bar{\pi}_i(t) \leq \bar{x}_i(t).
\end{aligned} \tag{8}$$

Let $\bar{V}_0^*(x)$ denote the optimal cost of (8).

For every time epoch t, we also define the fluid relaxation of the IP (6) as

$$\begin{aligned}
\max \ & \bar{R}(z) := \sum_{i=1}^{I} \frac{c_i \mu_i}{m_i} z_i + \Gamma \sum_{i=1}^{I} z_i \\
\text{s.t.} \quad & \sum_{i=1}^{I} z_i \leq N \\
& 0 \leq z_i \leq x_i m_i, \quad i = 1, \dots, I.
\end{aligned} \tag{9}$$

Note that (9) is a linear program (LP) relaxation of (6), i.e., without the integer constraints. It is also straightforward to see that, for any $\Gamma \geq 0$, the optimal solution to (9) is to prioritize according to the $c\mu/m$ index. In particular, let $\bar{z}^*$ denote the optimal solution to (9). We also denote $[i]$ as the i-th class in the decreasing order of the $c\mu/m$-index, i.e., $c_{[i]}\mu_{[i]}/m_{[i]} > c_{[i+1]}\mu_{[i+1]}/m_{[i+1]}$. Then

$$\bar{z}^*_{[j]} = \left(N - \sum_{i=1}^{j-1} \bar{z}^*_{[i]} \right)^+ \wedge x_{[j]} m_{[j]}, \quad j = 1, \ldots, I. \tag{10}$$

With a little abuse of terminology, we refer to the policy characterized by (9) as the fluid $c\mu/m$ rule.

LEMMA 2. *For the fluid cost minimization problem (8), it is optimal to follow the fluid $c\mu/m$ rule.*

THEOREM 3. *For any sequence of policies, π^η, if $x^\eta/\eta \to x$ as $\eta \to \infty$, then*

$$\liminf_{\eta\to\infty} \frac{1}{\eta} V_0^{\pi^\eta,\eta}(x^\eta) \geq \bar{V}_0^*(x). \tag{11}$$

For any fixed $\Gamma \geq 0$, if $x^\eta/\eta \to x$ as $\eta \to \infty$, then the sequence of idle-aware $c\mu/m$ rule satisfies

$$\lim_{\eta\to\infty} \frac{1}{\eta} V_0^{\pi^{IP(\Gamma),\eta},\eta}(x^\eta) = \bar{V}_0^*(x). \tag{12}$$

Theorem 3 indicates that $\pi^{\mathrm{IP}(\Gamma),\eta}$ is asymptotically optimal. The proof of the theorem is provided in Appendix C.3. Since the result in Theorem 3 holds for any $\Gamma \geq 0$, setting $\Gamma > N\sum_{i=1}^I c_i\mu_i/m_i$, we have the following corollary.

COROLLARY 1. *The idle-avoid $c\mu/m$ rule is asymptotically optimal to the MDP in (2), in the many-server regime.*

We note from Theorem 3 that when the system size is large, the performance of any idle-aware $c\mu/m$ rule are asymptotically indistinguishable. However, as seen in Section 3, for small systems, there can be significant differences in the performance of different idle-aware $c\mu/m$ rules. In particular, while the priority-induced idleness becomes negligible for large systems, it has a critical impact on performance in small systems.

6. Throughput Optimality of Idle-Avoid $c\mu/m$ Rule

In this section, we move away from the main problem setup in (2), and study the long-time behavior of the idle-avoid $c\mu/m$ rule. In contrast to the analysis in Section 5, we fix the size/scale of the system, and study its performance as $t \to \infty$. Analyzing long-time behavior requires us to impose more restrictions on the system parameters. In particular, we need some notion of long-run regularity of the arrival rates. In this section, we make the following assumption on the arrival probabilities.

ASSUMPTION 1. *There exists* $\bar{\lambda}_i \in [0,1]$, $i = 1, \ldots, I$, *such that*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} \lambda_i(s) = \bar{\lambda}_i.$$

In what follows, we refer to $\bar{\lambda} = (\bar{\lambda}_1, \ldots, \bar{\lambda}_I)$ as the limiting arrival rate.

The setting we studied in this section is quite different from the transient optimal scheduling problem we started with, but provides important insights into the performance of the proposed scheduling policies over relatively long time horizons.

When planning for a long time horizon, the first order goal is to ensure system stability, so that the queue will not grow without bound as time increases. We employ the notion of rate stability as in Armony and Bambos (2003).

DEFINITION 2. We define a System X (under a scheduling policy π) to be rate stable if

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{I} A_i(t) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{I} D_i(t) \text{ almost surely.}$$

We first note that due to the multiple resource requirements, $\rho := \sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{N\mu_i} < 1$ does not imply that the system can be stabilized. For example, in a single-class queue with $N = 3$ and $m = 2$, if $\bar{\lambda} \in (\mu, 3/2\mu)$, $\rho < 1$ but the system cannot be stabilized. Thus, we start by defining the maximum stability region of the system, i.e., the set of the arrival rates that can be processed/stabilized using some scheduling rule. For a multi-class system with different resource requirements, let $\phi^1, \ldots, \phi^K$ denote the list of all possible service configurations. In particular, $\phi^k = (\phi_1^k, \ldots, \phi_I^k)$, $k = 1, \ldots, K$, is a server allocation scheme satisfying the following conditions

$$\frac{\phi_i^k}{m_i} \in \mathbb{N}_0, \quad \sum_{i=1}^{I} \phi_i^k \le N.$$

Then, the maximum stability region of the system can be characterized by

$$\mathcal{M} = \left\{ \bar{\lambda} \in [0,1]^I : \bar{\lambda}_i m_i/\mu_i \le \sum_{k=1}^{K} \alpha_k \phi_i^k, \text{ for some } \alpha_k \ge 0,\ k = 1,\dots,K \text{ and } \sum_{k=1}^{K} \alpha_k = 1 \right\}.$$

In the above definition of $\mathcal{M}$, α_k's can be interpreted as the proportion of time service configuration ϕ^k is employed. A scheduling policy that achieves the maximum stability region is known to be throughput optimal. That is for any limiting arrival rate $\bar{\lambda} \in \mathbb{S}$, the scheduling policy can achieve rate stability.

Under the preemption and Markovian assumptions, any scheduling policy can be viewed as a mapping from the state of the system to an allocation of servers. Thus, we define the set of feasible scheduling policies as

$$\Omega = \left\{ \psi : \sum_{i=1}^{I} \psi_i(x) \le N,\ \ \psi_i(x)/m_i \le x_i,\ \ \psi_i(x)/m_i \in \mathbb{N}_0, \quad \text{for } i = 1,\dots,I, \text{ and } \forall x \in \mathbb{N}_0^I \right\}.$$

Note that not all the service configurations are feasible for a given state x, because we have the extra constraint that $\phi_i^k/m_i \le x_i$. We also define a special subset of feasible policies that minimizes the idleness in the system:

$$\Omega_m = \left\{ \psi : \psi \in \Omega,\ \ \psi(x) \in \arg\max_{\tilde{\psi} \in \Omega} \left\{ \sum_{i=1}^{I} \tilde{\psi}_i(x) \right\}, \forall x \in \mathbb{N}_0^I \right\}.$$

Note that the idle-avoid $c\mu/m$ rule belongs to Ω_m, while the other idle-aware $c\mu/m$ rules may not belong to Ω_m. The following theorem establishes that policies in Ω_m are throughput optimal under the assumption that the resource requirement has a hierarchical structure as defined in Assumption 2.

ASSUMPTION 2. *$m_i/m_{i+1} \in \mathbb{N}$ for $i = 1,\dots,I-1$, and $N/m_1 \in \mathbb{N}$, where $\mathbb{N}$ is the set of positive integers.*

THEOREM 4. *Under Assumptions 1 and 2, for any $\psi \in \Omega_m$, i.e., $\pi_i(t) = \psi(X(t))/m_i$, for $i = 1,\dots,I$, $t \ge 0$, if the limiting arrival rate $\bar{\lambda} \in \mathcal{M}$, then*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(t) = 0 \text{ almost surely,}$$

which implies that the system is rate stable.

The proof of Theorem 4 can be found in Appendix D. Since the idle-avoid $c\mu/m$ rule belongs to Ω_m, we have the following corollary.

COROLLARY 2. *Under Assumptions 1 and 2, The idle-avoid $c\mu/m$ rule is throughput optimal.*

6.1. Stability under Other Idle-Aware $c\mu/m$ Rules

In this section, we demonstrate through numerical experiments that the other idle-aware $c\mu/m$ rules may not be throughput optimal. Consider the two-class queues with different resource requirements, Figures 7 and 8 plot $\mathbb{E}[X_1(t)+X_2(t)]$ for different values of t. In Figure 7, we study a system with $m=(2,1)$ and $c_1\mu_1 < c_2\mu_2$. In this case, the $c\mu/m$ rule follows policy P_2, which may incur some idleness, while the idle-avoid $c\mu/m$ rule follows policy P_2^I. We note that under $\mathbf{P_2^I}$, $\mathbb{E}[X_1(t)+X_2(t)]$ is around 20 for all values of t. Under $\mathbf{P_2}$, $\mathbb{E}[X_1(t)+X_2(t)]$ is growing in t, suggesting that the system is not stable. In Figure 8, we study a system with $m=(3,1)$ and $c_1\mu_1 < c_2\mu_2$. In this case, we have three idle-aware $c\mu/m$ rules depending on the value of Γ. $\Gamma=0$ leads to P_2, which is the $c\mu/m$ rule, $\Gamma=0.25$ leads to P_2^{I1}, while $\Gamma=100$ leads to P_2^{I2}, which is the idle-avoid $c\mu/m$ rule. We note that only $\mathbf{P_2^{I2}}$ stabilizes the system. Under either $\mathbf{P_2}$ or $\mathbf{P_2^{I1}}$, $\mathbb{E}[X_1(t)+X_2(t)]$ is increasing in t. As $\mathbf{P_2}$ can incur more idleness than $\mathbf{P_2^{I1}}$, $\mathbb{E}[X_1(t)+X_2(t)]$ increases faster in t under $\mathbf{P_2}$ than under $\mathbf{P_2^{I1}}$.

Figure 7 $\mathbb{E}[X_1(t)+X_2(t)]$ as a function of t under different idle-aware $c\mu/m$ policies. $N=15$, $\lambda=(0.26,1)$, $\mu=(0.35,0.7)$, and $c=(1,0.8)$.

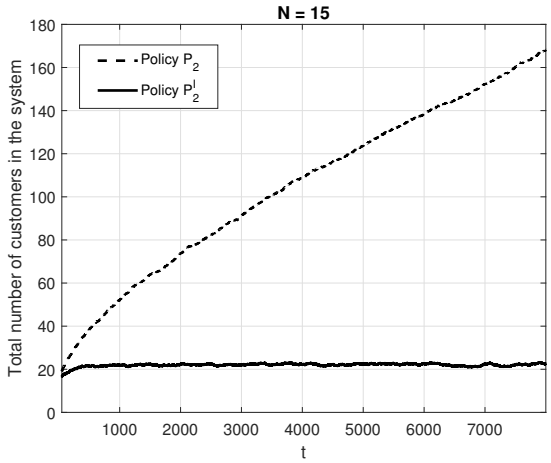
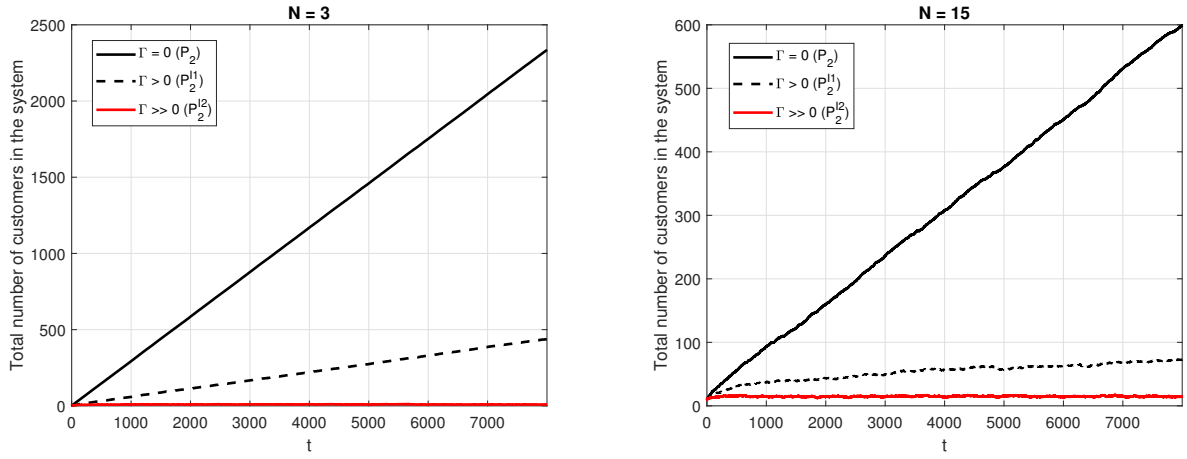


Figure 8 $\mathbb{E}[X_1(t)+X_2(t)]$ as a function of t under different idle-aware $c\mu/m$ policies. $I=2$, $N=3\eta$, $\eta=1$ (left plot) and $\eta=5$ (right plot), $m=(3,1)$, $A_i(t)\sim Binomial(\eta,\lambda_i)$, where $\lambda=(1/24,1/8)$, $\mu=(0.5,1)$, $c=(1,0.8)$ and $x(0)=(N/3,N-1)$.

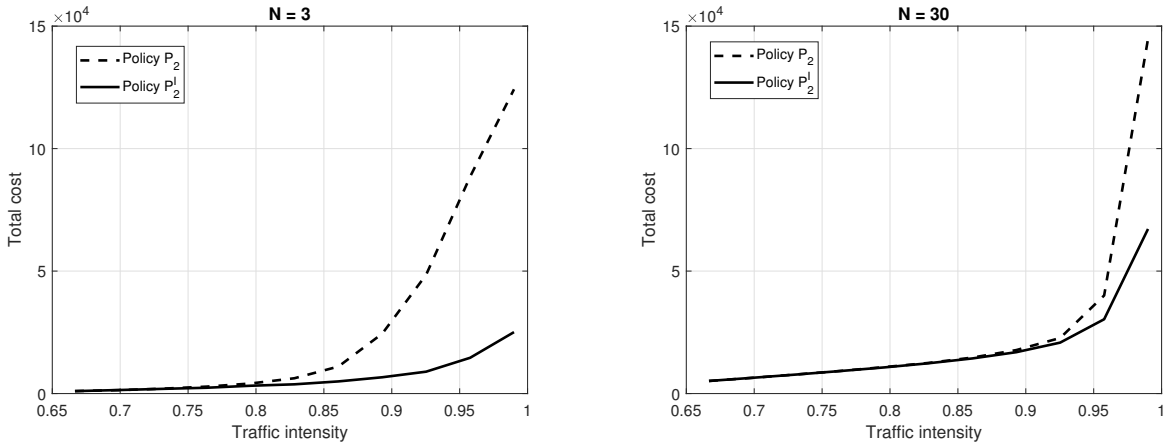


6.2. Implications for Holding Cost over Long Time Horizons

For a specific time-homogenous system, we call a policy stable if the system under this policy is rate stable, and we call a policy unstable otherwise. If a policy is unstable, the corresponding queue can grow without bound as time increases. This implies that when planning for long time horizons, the difference in holding cost between stable and unstable policies can be very large. In Figure 9, we consider a two-class queue with $m=(2,1)$ and compare the holding costs under $\mathbf{P_2}$ (the $c\mu/m$ rule) and $\mathbf{P_2^I}$ (the idle-avoid $c\mu/m$ rule) over a very long planning horizon, i.e., $T=2000$. We vary the value of ρ, which is defined in (7), by scaling down the service rates. Note that for small values of ρ, the performances of $\mathbf{P_2}$ and $\mathbf{P_2^I}$ are very similar. However, as ρ increases, $\mathbf{P_2}$ leads to a much higher cost than $\mathbf{P_2^I}$. We also note that as the system size increases from $N=3$ to $N=30$, significant differences in performance between $\mathbf{P_2}$ and $\mathbf{P_2^I}$ start occurring at a higher value of ρ. For example, for $N=3$, when $\rho=0.9$, the cost under $\mathbf{P_2}$ is more than twice the cost under $\mathbf{P_2^I}$. However, for $N=30$, when $\rho=0.9$, the costs under the two policies are almost the same.

In the next experiment, we fix the traffic intensity ρ, but vary the scale of the system. Figure 10 compares the costs of the $c\mu/m$ rule and the idle-avoid $c\mu/m$ rule over a very long planning horizon, i.e., $T=2000$. We consider the two-class queues with $c_1\mu_1<c_2\mu_2$. The systems have time-homogeneous arrival probabilities, which we sample from $U[0,1]$.

Figure 9 Total cost for P_2 versus P_2^I. $T = 2000$, $I = 2$, $m = (2,1)$, $N = 3\eta$, $\mu = (0.5/k, 1/k)$, k varies between 1, and 1.485, $c = (1, 0.8)$, $\xi = 5$, and $A_i(t) \sim Binomial(\eta, \lambda_i)$ with $\lambda = (0.25, 1)$. On the left, $\eta = 1$, and on the right, $\eta = 10$.



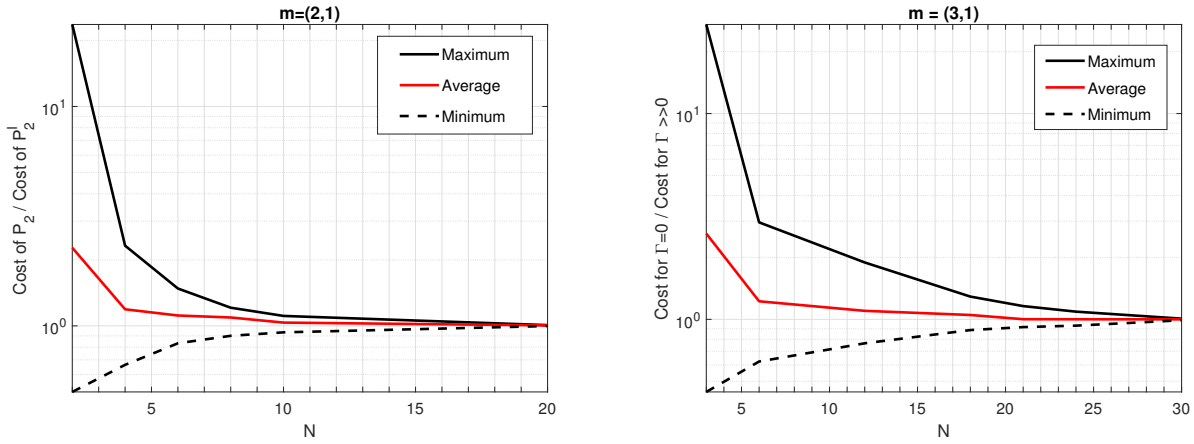
In the left plot $m = (2,1)$, we compare $\mathbf{P_2}$ versus $\mathbf{P_2^I}$. We plot the maximum, minimum, and average of $V_0^{\mathbf{P_2}}(x)/V_0^{\mathbf{P_2^I}}(x)$, with $x = (\lfloor N/2 \rfloor, N-1)$, over 50 randomly drawn problem instances (arrival probabilities). We observe that when $N = 2$, the maximum ratio between the two costs can be very large, i.e., the cost under $\mathbf{P_2}$ can be more than 11 times the cost under $\mathbf{P_2^I}$. On the other hand, the minimum ratio between the two policies is bounded by 1/2 as suggested by Theorem 2. This suggests that the idle-avoid $c\mu/m$ rule achieves more robust performance than the $c\mu$ rule when planning over a long time horizon, especially in small systems. As N increases, the performance of the two policies are practically indistinguishable as suggested by Theorem 3. In the right plot, $m = (3,1)$ and we compare costs under $\mathbf{P_2}$ versus $\mathbf{P_2^{I2}}$. We observe again that the idle-avoid $c\mu/m$ achieves more robust performance than the $c\mu/m$ rule.

The analysis in this section provides additional evidence supporting the use of the idle-avoid $c\mu/m$ rule in practice. This is especially important in small systems where priority-induced idleness can lead to very bad performance. We also emphasize that there is value to understanding small systems. In the healthcare setting, the number of servers (nurses/beds) in a unit is more commonly on the order of 10s instead of 100s or 1000s.

7. Additional Numerical Experiments

In this section, we provide additional numerical experiments to provide more insights into the performance of our proposed policy. We first look at preemption versus non-preemption.

Figure 10 Cost comparison (log) – $c\mu/m$ rule vs. idle-avoid $c\mu/m$ for different values of N. $T = 2000$, $I = 2$, $c = (1,1)$, and $\xi = 5$. On the left, Policy $\mathbf{P_2}$ vs. Policy $\mathbf{P_2^I}$ for $m = (2,1)$, $N = 2\eta$, $\eta = 1,\ldots,10$, $\mu = (0.25,1)$, $A_i(t) \sim Binomial(\eta,\lambda_i)$, where $\lambda_1,\lambda_2 \sim U[0,1]$, and $X(0) = (\lfloor N/2 \rfloor, N-1)$. On the right, cost ratio between $\mathbf{P_2}$ ($\Gamma = 0$) and $\mathbf{P_2^{I2}}$ ($\Gamma \gg 0$) for $m = (3,1)$, $N = 3\eta$, $\eta = 1,\ldots,10$, $\mu = (0.09,1)$, $A_i(t) \sim Binomial(\eta,\lambda_i)$, where $\lambda_1,\lambda_2 \sim U[0,1]$, and $X(0) = (\lfloor N/3 \rfloor, N-2)$.



Our theoretical analysis assumes that preemption is allowed. In Section 7.1, we investigate how the insights from our analysis for preemptive systems can be generalized to non-preemptive systems. We then compare the idle-avoid $c\mu/m$ rule to the max-weight policy in Section 7.2. We have shown that the idle-avoid $c\mu/m$ is throughput optimal in Section 6. Another important class of throughput-optimal policies is the max-weight policy. It is of interest to compare the performance of both policies.

7.1. Non-Preemption

Consider imposing non-preemption in the idle-aware $c\mu/m$ rules. In particular, we require that once a customer starts service, he/she cannot be moved back to the queue. This is natural in many service systems, and particularly healthcare systems. Let $x = (x_1,\ldots,x_I)$ denote the state of the system at the beginning of time epoch t, $z' = (z'_1,\ldots,z'_I)$ denote the number of servers occupied by each class before assignment, and $z = (z_1,\ldots,z_I)$ denote the number of servers allocated to each class after assignment. The new IP under non-

preemption is defined as

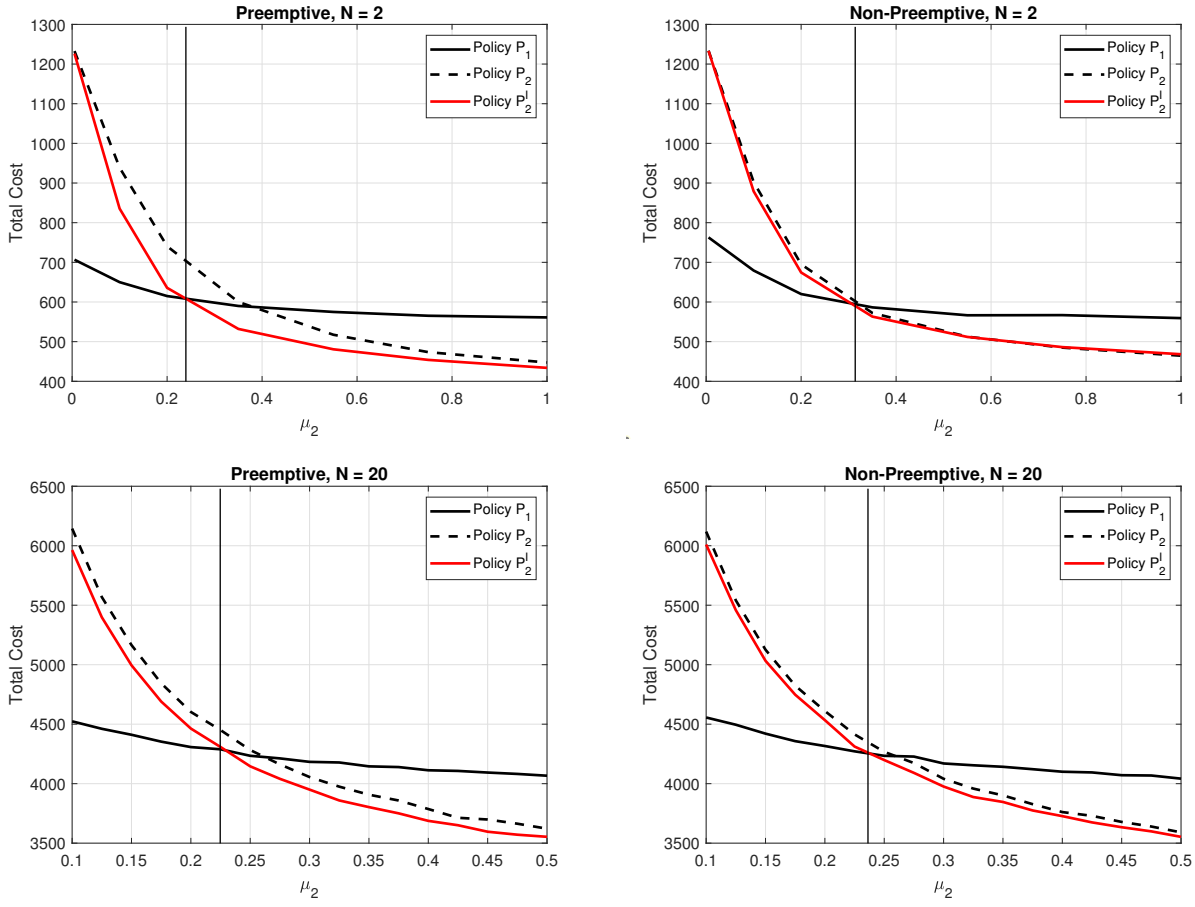
$$\begin{aligned} \max \ R_N(z) := & \sum_{i=1}^{I} \frac{c_i \mu_i}{m_i} z_i + \Gamma \sum_{i=1}^{I} z_i \\ \text{s.t.} \quad & \sum_{i=1}^{I} z_i \le N \\ & z_i' \le z_i \le x_i m_i, \quad i = 1, \ldots, I \\ & z_i / m_i \in \mathbb{N}_0, \quad i = 1, \ldots, I, \end{aligned}$$

In general, non-preemption can introduce a number of challenges. For example, it may be optimal to keep servers idle in anticipation of more 'important' incoming customers (Pinedo 2012). As such, we focus on comparison only across different non-preemptive idle-aware $c\mu/m$ rules.

We know that translation of results derived from a preemptive system to a non-preemptive system does not always result in good performance (see, e.g., Rozenshmidt (2008)). However, we expect that as the system size grows, the difference between non-preemption and preemption will be minimal (Atar et al. 2004). For example, we expect the idle-avoid $c\mu/m$-rule to perform well for large systems even without preemption.

Consider a two-class model with $m = (2, 1)$ and three possible scheduling policies according to the idle-aware $c\mu/m$ rules: $\mathbf{P_1}$, $\mathbf{P_2}$ and $\mathbf{P_2^I}$. Figure 11 compares $V_0^\pi(x)$ under the preemptive (left plots) versus non-preemptive (right plots) assumptions. The top plots are for a small system with only $N = 2$ servers, while the bottom plots are for $N = 20$. The vertical lines in the figures depict where the cost under $\mathbf{P_2^I}$ becomes smaller than the cost under $\mathbf{P_1}$ as μ_2 increases. Note that when preemption is allowed, it is where $c_1\mu_1/2 = c_2\mu_2$. We observe that even though the two systems have different costs, in both systems, the optimal policy among the three policies considered, switches from prioritizing Class 1 to prioritizing Class 2 as μ_2 increases. The value of μ_2 at which $\mathbf{P_2^I}$ surpasses $\mathbf{P_1}$, i.e., the vertical line, is different in the two systems. In the preemptive system, it is at $\mu_2 = 0.225$. In the non-preemptive system, it is at $\mu_2 = 0.33$. For sufficiently large values of μ_2, $\mathbf{P_2}$ can perform better than $\mathbf{P_2^I}$, but the difference is very small. In the bottom plots, the system size, $N = 20$, can reasonably capture the size of an average ICU. We observe that in this case, non-preemption does not lead to much cost difference. When comparing the three policies, the optimal policy switches from $\mathbf{P_1}$ to $\mathbf{P_2^I}$ as μ_2 increases. $\mathbf{P_2^I}$ leads to slightly better performance than $\mathbf{P_2}$ in both the preemptive and nonpreemptive cases.

Figure 11 Cost comparison – preemption (left plot) vs. non-preemption (right plot) for $N = 2\eta$, $\eta = 1$ (top plots) and $\eta = 10$ (bottom plots), $I = 2$, $m = (2, 1)$, and different values of μ_2. Here, $T = 50$, $A_i(t) \sim Binomial(\eta, \lambda_i(t))$, $\lambda(t) = (0.05, 0.15)$, for all t, $\mu_1 = 0.09$, $c = (5, 1)$, $\xi = 5$, and $X(0) = (N, N)$.



Note that we can also come up with extreme examples in small systems where policy $\mathbf{P_2^I}$ can perform much worse than $\mathbf{P_2}$ when preemption is not allowed. For example, consider a system where $N = 2$ and the service rate of Class 1 is very small. Policy $\mathbf{P_2^I}$ may admit a Class 1 customer over a Class 2 customer at some point. Then, the Class 1 customer will "block" the servers for a very long time. Meanwhile, policy $\mathbf{P_2}$ can keep admitting Class 2 customers. If the service rate of Class 2 customers is sufficiently larger, $\mathbf{P_2}$ can achieve a much lower cost than $\mathbf{P_2^I}$ in this case. Thus, one must be careful when operating small and nonpreemptive systems in these types of extreme parameter regimes.

7.2. Comparison with the Max-Weight Policies

We compare the idle-avoid $c\mu/m$ rule to the max-weight policy, which is also known to be throughput optimal (Stolyar 2004, Armony and Bambos 2003). Note that the idle-avoid $c\mu/m$ rule is primarily designed to handle linear holding cost and transient cost minimization problems with arbitrary time-varying arrival rates, while the optimality of the max-weight policy with respect to cost minimization requires strongly convex holding costs and is in a long time-horizon sense (see Stolyar (2004)). To facilitate a relatively fair comparison, we consider time-homogeneous arrival probabilities and look at longer time horizons. At each time t, given $X(t) = x$, the server allocation, $z = (z_1, \ldots, z_I)$, under the max-weight policy is the solution to the following IP:

$$\begin{aligned} &\max_z \sum_{i=1}^{I} \frac{c_i \mu_i}{m_i} z_i x_i^{\beta} \\ &\text{s.t.} \quad \sum_{i=1}^{I} z_i \le N, \quad 0 \le \frac{z_i}{m_i} \le x_i, \frac{z_i}{m_i} \in \mathbb{N}_0, \ i = 1, \ldots, I, \end{aligned} \tag{13}$$

We allow different values of $\beta > 0$. $\beta = 1$ is commonly used in the literature (Armony and Bambos 2003, Dai and Lin 2005). When β is small, the convex cost function, i.e., $\sum_{i=1}^{I} c_i X_i^{\beta+1}$, is 'close' to being linear; thus, this max-weight policy with β close to 0 should have performance that is close to optimal for our linear objective function (Stolyar 2004).

Figure 12 plots the ratio of the total cost of the max-weight policy to that of the idle-avoid $c\mu/m$ rule for different values of μ_2 in the regime where $c_1\mu_1 < c_2\mu_2$. We consider two values of β for the max-weight policy: 1 and 0.1. We observe that the max-weight policies have worse performance than the idle-avoid $c\mu/m$ rule. When $\beta = 1$, the cost ratio can be above 1.5 for small values of μ_2 and is slightly above 1 for large values of μ_2. On the other hand, when $\beta = 0.1$, the cost ratio can be above 4 for large values of μ_2 and is slightly above 1 for small values of μ_2. To take a closer look at the reason behind the poor performance of the max-weight policy with $\beta = 0.1$ when μ_2 is large, in Figure 13, we plot the average queue length process (averaged over 50 sample paths) under the three policies when $\mu_2 = 0.85$. While all three policies stabilize the system, the number of customers in each class are quite different. When $\beta = 0.1$, the max-weight policy prioritizes Class 2 customers too much, which can cause a significant amount of priority-induced idleness. This leads to a very small Class 2 queue but an extremely large Class 1 queue.

Figure 12 Total cost incurred by each policy for different values of μ_2. $T = 5000$, $N = 2$, $I = 2$, $m = (2,1)$, $\lambda(t) = (0.2, 0.32)$, for all t, $\mu_1 = 0.3$, $c = (1,1)$, $\xi = 5$, and $\Gamma = 100$.

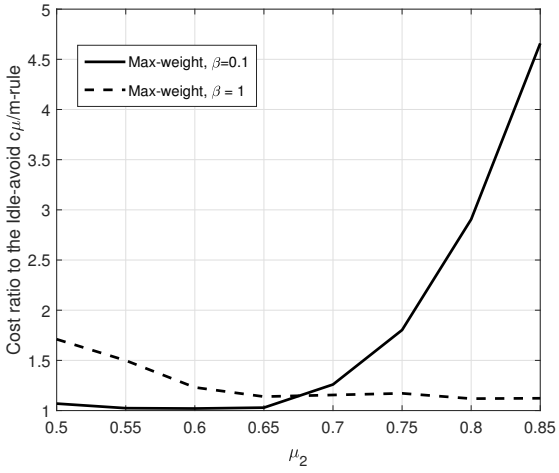
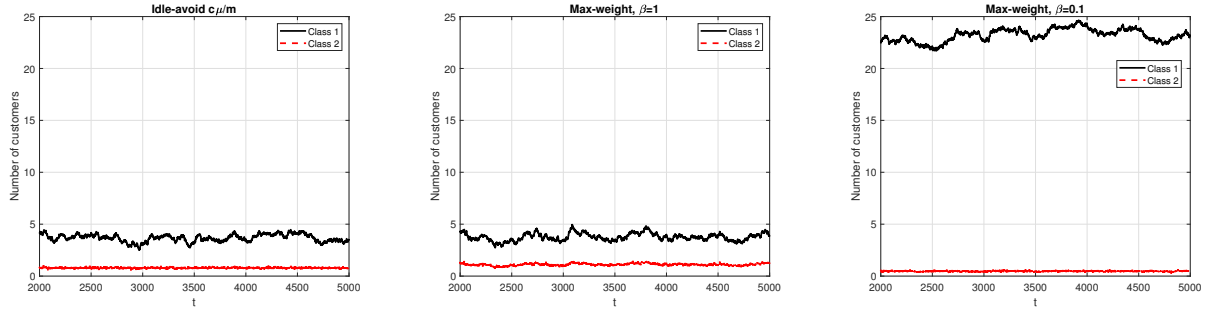


Figure 13 Total number of customers as a function of t under different policies. $N = 2$, $I = 2$, $m = (2,1)$, $\lambda(t) = (0.2, 0.32)$, for all t, $\mu = (0.3, 0.85)$, $\xi = 5$, $c = (1,1)$ and $\Gamma = 100$.



To conclude this section, we would like to point out that in order to have a fair comparison, we do not show experiments with large initial values, time-varying arrivals, or short time horizons. In these cases, the idle-avoid $c\mu/m$ rule can significantly outperform the max-weight policy. This is not surprising since the idle-avoid $c\mu/m$ rule is designed for such problems, while the max-weight policy is designed for very long time horizons, even though it is agnostic to the arrival rates.

8. Discussion and Future Directions

In this paper, we study the optimal scheduling policy for multi-server queues with multiple classes of customers. The special feature we study is that different classes of customers may require a different number of servers. We propose an index-based policy, the idle-avoid

$c\mu/m$ rule, that minimizes the amount of idleness incurred while prioritizing customers according to their $c\mu/m$ index. We prove that this policy is asymptotically optimal under the many-server regime and is throughput optimal under certain regularity conditions on the arrival probabilities.

We find that the addition of the different resource requirements introduces new dynamics that did not arise in the classical multi-class queues. In particular, the impact of priority-induced idleness is a direct consequence of the different resource requirements. As we have shown, this idleness can have substantial consequences, such as leading to poor performance of seemingly good policies. While avoiding idleness may sacrifice the instantaneous cost reduction rate, we demonstrate theoretically and through numerical experiments that the amount of suboptimality is limited. More specifically, our results indicate that in small and heavily loaded systems, which are highly relevant to healthcare applications, avoiding idleness is crucial. When the system is very large and/or very lightly loaded, there is more slack in capacity to accommodate idleness; hence, other idle-aware $c\mu/m$ rules, such as the $c\mu/m$ rule, can perform well. However, because uncertainty in demand (e.g., unpredictable disease outbreaks) can quickly alter the system dynamics, we recommend using the idle-avoid $c\mu/m$ rule all the time unless the system administrator is certain that the system is lightly loaded, in which case, the holding costs are likely to be low anyway.

We identify several directions for future research from the modeling perspective. First, it would be interesting to study a network of resources instead of a single type of resource. For example, in the ICU setting, we need both an ICU bed and the required nurses to admit a patient, and either one can be the bottleneck. The challenge then is to develop good scheduling policies that balance multiple resource constraints. Second, for ICUs in particular, patients' acuity levels may evolve over time, suggesting that the same patient may have different resource requirements during his/her length of stay. The staffing level can also change from shift to shift. It would be interesting to incorporate these time-varying dynamics (e.g., class transition behavior) and study the acuity-based optimal staffing policy in this setting.

Acknowledgement

The authors thank the area editors Itai Gurvich and Amy Ward, an anonymous associate editor, and two anonymous reviewers for many valuable comments and suggests that

greatly helped improve the paper. Jing Dong was supported in part by a National Science Foundation grant CMMI-1762544. Noa Zychlinski was supported in part by the Eric and Wendy Schmidt Postdoctoral Award for Women in Mathematical and Computing Sciences, and the Israeli Council for Higher Education.

Author Biographies

Noa Zychlinski is an Assistant Professor in the Faculty of Industrial Engineering and Management at the Technion – Israel Institute of Technology. Noa was a postdoctoral research fellow in the Division of Decision, Risk and Operations at Columbia Business School. Her research interests focus on service and healthcare operations management, stochastic modeling, queueing theory applications, and stochastic process approximation.

Carri W. Chan is a Professor in the Division of Decision, Risk, and Operations at Columbia Business School. Her primary research interests are in data-driven modeling of complex stochastic systems, dynamic optimization, and queueing with applications in health-care operations management. Her current focus is on combining empirical approaches with mathematical modeling to develop evidence-based approaches to improving patient flow through hospitals, and particularly intensive care units.

Jing Dong Jing Dong is an Associate Professor in the Decision, Risk, and Operations division at the Graduate School of Business, Columbia University. Her primary research interests are in applied probability and stochastic simulation, with an emphasis on applications in service operations management. Her current research focuses on developing data-driven stochastic modeling to improve patient flow in hospitals.

References

Altay, N. 2012. Capability-based resource allocation for effective disaster response. *IMA Journal of Management Mathematics* **24**(2) 253–266. 2

Armony, M., N. Bambos. 2003. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems* **44**(3) 209–252. 6, 25, 33, 52, 53

Armony, M., S. Israelit, A. Mandelbaum, Y. Marmor, Y. Tseytlin, G. Yom-Tov. 2015. On patient flow in hospitals: A data-based queueing-science perspective. *Stochastic Systems, November* **6**(3). 5

Atar, R., C. Giat, N. Shimkin. 2010. The cμ/θ rule for many-server queues with abandonment. *Operations Research* **58**(5) 1427–1439. 6

Atar, R., A. Mandelbaum, M.I. Reiman. 2004. Scheduling a multi class queue with many exponential servers: Asymptotic optimality in heavy traffic. *The Annals of Applied Probability* **14**(3) 1084–1134. 6, 31

Baron, O., O. Berman, D. Krass, J. Wang. 2014. Using strategic idleness to improve customer service experience in service networks. *Operations Research* **62**(1) 123–140. 5

Baron, O., O. Berman, D. Krass, J. Wang. 2017. Strategic idleness and dynamic scheduling in an open-shop service network: Case study and analysis. *Manufacturing & Service Operations Management* **19**(1) 52–71. 5

Bramson, M. 1994. Instability of FIFO queueing networks. *Annals of applied probability* **4** 414–431. 6

Brilli, R.J., A. Spevetz, R.D. Branson, G.M. Campbell, H. Cohen, J.F. Dasta, M.A. Harvey, M.A. Kelley, K.M. Kelly, M.I. Rudis. 2001. Critical care delivery in the intensive care unit: defining clinical roles and the best practice model. *Critical care medicine* **29**(10) 2007–2019. 2

Buyukkoc, C., P. Varaiya, J. Walrand. 1985. The cμ rule revisited. *Advances in applied probability* **17**(1) 237–238. 19

Carayon, P., A.P. Gurses. 2008. Nursing workload and patient safety—a human factors engineering perspective. *Patient Safety and Quality: An Evidence-Based Handbook for Nurses*. Rockville, MD: Agency for Healthcare Research and Quality (US), 1–14. 2

Chan, C. W., L. V. Green, S. Lekwijit, L. Lu, G. Escobar. 2018. Assessing the impact of service level when customer needs are uncertain: An empirical investigation of hospital step-down units. *Management Science* **65**(2) 751–775. 2

Cox, D.R., W.L. Smith. 1961. Queues. *Methuen, London* . 5

Dai, J.G., W. Lin. 2005. Maximum pressure policies in stochastic processing networks. *Operations Research* **53**(2) 197–218. 6, 33

Fachruddin, N., W. Santoso, A. Zakiyah. 2019. The relationship between workload with work stress on nurses in intensive installation of bangil general hospital pasuran district. *International Journal of Nursing and Midwifery Science (IJNMS)* **2**(03) 311–321. 2

Gans, N., G. van Ryzin. 1997. Optimal control of a multiclass, flexible queueing system. *Operations Research* **45**(5) 677–693. 5

Green, L. 1980. A queueing system in which customers require a random number of servers. *Operations research* **28**(6) 1335–1346. 2, 4

Green, L. 1981. Comparing operating characteristics of queues in which customers require a random number of servers. *Management Science* **27**(1) 65–74. 4

Gurvich, I., J.A. Van Mieghem. 2017. Collaboration and multitasking in networks: Prioritization and achievable capacity. *Management Science* **64**(5) 2390–2406. 6

Harrison, J.M. 1998. Heavy traffic analysis of a system with parallel servers: asymptotic optimality of discrete-review policies. *Annals of applied probability* 822–848. 6

Harrison, J.M., A. Zeevi. 2004. Dynamic scheduling of a multiclass queue in the halfin-whitt heavy traffic regime. *Operations Research* **52**(2) 243–257. 6

Huang, J., B. Carmeli, A. Mandelbaum. 2015. Control of patient flow in emergency departments, or multiclass queues with deadlines and feedback. *Operations Research* **63**(4) 892–908. 5

Kim, J., R. Randhawa, A. Ward. 2018. Dynamic scheduling in a many-server, multiclass system: The role of customer impatience in large systems. *Manufacturing and Service Operations Management* **20**(2) 285–301. 6

Kim, S-H., E.j. Pinker, J. Rimar, E.H. Bradley. 2017. Refining workload measure in hospital units: From census to acuityadjusted census in intensive care units. Tech. rep., Working paper, USC, Marshall School of Business, Los Angeles, CA. 2

Luo, J., J. Zhang. 2013. Staffing and control of instant messaging contact centers. *Operations Research* **61**(2) 328–343. 2

Mandelbaum, A., A.L. Stolyar. 2004. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research* **52**(6) 836–855. 5

Masterson, G., S. Baudouin. 2015. Guidelines for the provision of intensive care services. *London: Faculty of Intensive Care Medicine* . 2

Mills, A.F., N.T. Argon, S. Ziya. 2013. Resource-based patient prioritization in mass-casualty incidents. *Manufacturing & Service Operations Management* **15**(3) 361–377. 5

Mueller, M., S. Lohmann, R. Strobl, C. Boldt, E. Grill. 2010. Patients' functioning as predictor of nursing workload in acute hospital units providing rehabilitation care: a multi-centre cohort study. *BMC health services research* **10**(1) 295. 2

O'Brien-Pallas, L., D. Irvine, E. Peereboom, M. Murray. 1997. Measuring nursing workload: understanding the variability. *Nursing Economics* **15**(4) 171–183. 2

Papadimitriou, C.H., J.N. Tsitsiklis. 1999. The complexity of optimal queuing network control. *Mathematics of Operations Research* **24**(2) 293–305. 19

Pinedo, Michael. 2012. *Scheduling*. Springer. 31

Puha, A.L., A.R. Ward. 2019. Scheduling an overloaded multiclass many-server queue with impatient customers. *Operations Research & Management Science in the Age of Analytics*. INFORMS, 189–217. 6

Puterman, M.L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, Hoboken, New Jersey. 9

Reiman, M.I. 1991. A critically loaded multiclass erlang loss system. *Queueing Systems* **9**(1-2) 65–81. 5

Rozenshmidt, Lubov. 2008. *On priority queues with impatient customers: Stationary and time-varying analysis*. Technion-Israel Institute of Technology, Faculty of Industrial and Management. 31

Rybko, A.N., A.L. Stolyar. 1992. Ergodicity of stochastic processes describing the operations of open queueing networks. *Problems Inform. Transmission* **28** 119–220. 6

Saghafian, S., W.J. Hopp, M.P. Van Oyen, J.S Desmond, S.L. Kronick. 2014. Complexity-augmented triage: A tool for improving patient safety and operational efficiency. *Manufacturing & Service Operations Management* **16**(3) 329–345. 5

Sherali, H.D., T.B. Carter, A.G. Hobeika. 1991. A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B: Methodological* **25**(6) 439–452. 2

Stolyar, A. 2004. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability* **14**(1) 1–53. 6, 33

Sun, Z., N. T. Argon, S. Ziya. 2018. Patient triage and prioritization under austere conditions. *Management Science* **64**. 5

Tarnow-Mordi, W.O., C. Hau, A. Warden, A.J. Shearer. 2000. Hospital mortality in relation to staff workload: A 4-year study in an adult intensive-care unit. *The Lancet* **356**(9225) 185–189. 2

Van Mieghem, J.A. 1995. Dynamic scheduling with convex delay costs: The generalized cμ rule. *The Annals of Applied Probability* 809–833. 5

Yankovic, N., L.V. Green. 2011. Identifying good nursing levels: A queuing approach. *Operations research* **59**(4) 942–955. 5

Appendix A: Proof of Theorem 1 - Cases 2a and 2b

Recall that $\hat{\pi}$ denotes the policy characterized by Theorem 1 and π^* denotes the optimal policy.

A.1. Case 2a. $\hat{\pi} = \mathbf{P_2}$

Base Case: $t = T-1$. Due to the linearity of (4) in $\pi(t)$, it is straightforward to see that when $c_2\mu_2 > c_1\mu_1$, we shall allocate as much capacity as possible to Class 2. Thus, $\pi^*(T-1) = P_2 = \hat{\pi}(T-1)$.

Inductive step. Suppose that $\hat{\pi}$ minimizes V_t^{π} for some $1 \le t \le T-1$. We show by contradiction that at time $t-1$, it is also optimal to follow $\hat{\pi}$. Suppose by contradiction that at time $t-1$, it is optimal to follow some other policy, and then by the induction argument, from time t onward, we shall follow $\hat{\pi}$.

We consider two coupled systems s^* and s'. Both systems start from the same state at $t-1$, i.e., $X(t-1) = x$, and see the same arriving customers. s^* system uses policy π^* while the s' system uses a (possibly) suboptimal policy π' that will be specified later.

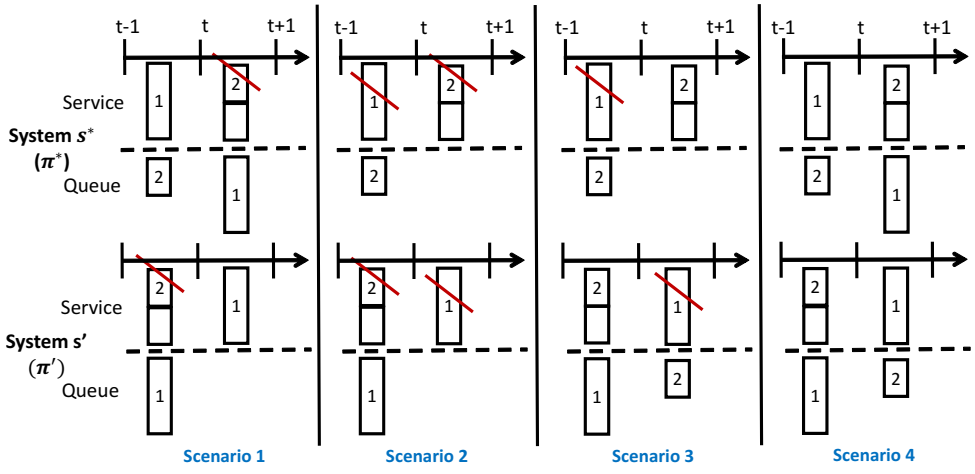
We conduct the analysis for different values of x.

- **If $x_1 = 0$ or $x_2 = 0$**, π^* and $\hat{\pi}$ coincide.
- **If $x_1 \ge 1$, $x_2 = 1$**, for π^* and $\hat{\pi}$ to deviate, π^* should admit one Class 1 customer at $t-1$.

For π', the Class 2 customer is admitted at $t-1$. There are four potential outcomes across the two systems at the end of time epoch $t-1$ (see Figure 14 for a pictorial illustration):

1. Only the Class 2 customer completes service.
2. Both the Class 1 customer and the Class 2 customer complete their service.
3. Only the Class 1 customer completes service.
4. Neither customer completes service.

Figure 14 Coupling illustration for Case 2a, Scenarios 1–4.



We construct policy π' such that we will admit the Class 1 customer at time t, and from time $t+1$ onward, π' will follow π^*. Under the coupling construction, the two systems, s^* and s', are fully synchronized at $t+1$ under each of the 4 scenarios. Thus, the cost difference between s^* and s' is the difference in the holding costs incurred at t. The cost difference (ΔC_t) and the corresponding probability (Pr) for each scenario is summarized in Table 3. Putting all the scenarios together, we have

$$V_{t-1}^{\pi^*}(X) - V_{t-1}^{\pi'}(X) = c_2(1-\mu_1)\mu_2 + \mu_1\mu_2(c_2-c_1) - c_1\mu_1(1-\mu_2) = \mu_2 c_2 - \mu_1 c_1 > 0,$$

where the last inequality comes from condition of Case 2a. This contradicts the assumption that π^* is optimal.

Table 3 The cost difference and the corresponding probability for each scenario

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
ΔC_t	c_2	$c_2 - c_1$	$-c_1$	0
Pr	$(1-\mu_1)\mu_2$	$\mu_1\mu_2$	$\mu_1(1-\mu_2)$	$(1-\mu_1)(1-\mu_2)$

- **If $x_1 \geq 1$ and $x_2 \geq 2$**, following similar lines of argument as in the case where $x_1 \geq 1$ and $x_2 = 2$, we can also show that admitting two Class 2 customers is preferable over admitting one Class 1 customer, i.e., $\hat{\pi}$ is optimal. Note that we have already proved that admitting one Class 2 customer is preferable over admitting one Class 1 customer, and admitting two Class 2 customers is straightforwardly preferable over admitting one Class 2 customer.

A.2. Case 2b. $\hat{\pi} = \mathbf{P_2^I}$

Base case: $t = T-1$. Due to the linearity of (4) in $\pi(t)$ and since $2c_2\mu_2 > c_1\mu_1$, we would prefer admitting two Class 2 customers over one Class 1 customer. However, since $c_2\mu_2 < c_1\mu_1$, we would prefer admitting one Class 1 customer over admitting only one Class 2 customer. Thus, $\pi^*(T-1) = P_2^I = \hat{\pi}(T-1)$.

Inductive step. Suppose $\hat{\pi}$ minimizes V_t^π for some $1 \leq t \leq T-1$. We show by contradiction that at time $t-1$, it is also optimal to follow $\hat{\pi}$. Suppose by contradiction that at time $t-1$, it is optimal to follow some other policy, and then by the induction argument, from time t onward, we shall follow $\hat{\pi}$.

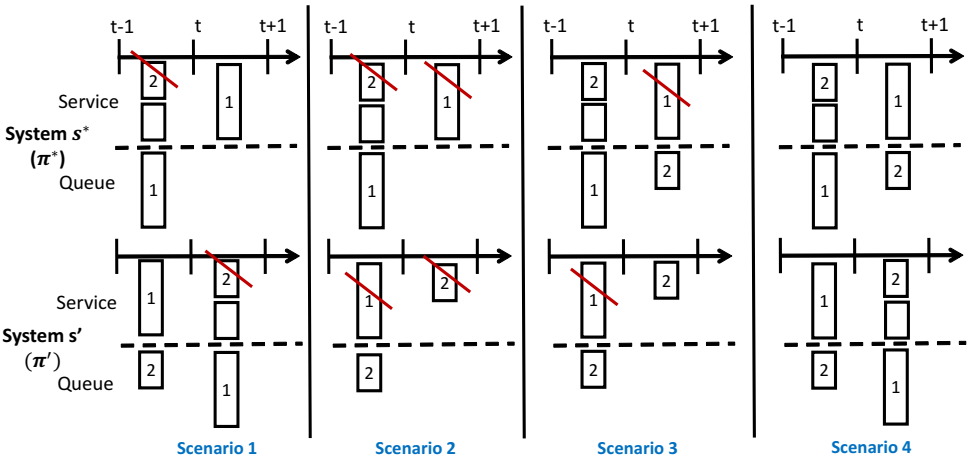
We consider two coupled systems s^* and s'. Both systems start in the same state at $t-1$, i.e., $X(t-1) = x$, and see the same arriving customers. The s^* system uses policy π^* while the s' system uses a suboptimal policy π' that will be specified later.

We next conduct the analysis for different values of x.

- **If $x_1 = 0$ or $x_2 = 0$,** the π^* and $\hat{\pi}$ do not deviate.
- **If $x_1 \geq 1$ and $x_2 = 1$,** for π^* to deviate from $\hat{\pi}$, in System s^*, we admit the Class 2 customer at $t-1$. In System s', we admit one Class 1 customer at $t-1$. Similar to Case 2a, there are four possible outcomes across the two systems. Their corresponding probabilities are also the same as in Case 2a.

If there is no Class 2 arrival at $t-1$, we construct π' such that we admit the Class 2 customer at t, and from $t+1$ onward, π' follows $\hat{\pi}$. From Figure 15, it is easy to see that the two systems are fully synchronized at $t+1$ in all scenarios.

Figure 15 Coupling illustration for Case 2b with one Class 2 customer and no Class 2 arrival at $t-1$.



If there is a Class 2 arrival at $t-1$, π' admits two Class 2 customers at t. In this case, we will allocate an additional server to s^* at t. We denote this new system with the extra server at t by $\tilde{s}$ and the corresponding optimal policy by $\tilde{\pi}$. Let $G_t(x)$ denote the optimal cost to go with $X(t) = x$ when having an extra server in time slot t only. From Lemma 3 in Appendix A.3, we have

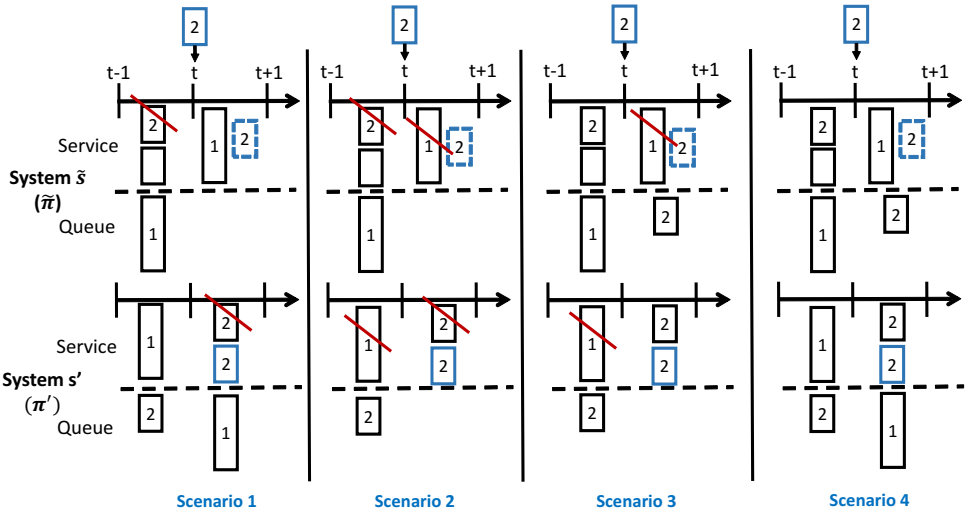
$$G_t(S_{t-1}(x, \pi^*(t-1))) \leq V_t^{\pi^*}(S_{t-1}(x, \pi^*(t-1))).$$

From time $t+1$ onward, both systems follow the same policy, i.e., π^*. Figure 16 provides a pictorial illustration of the coupling. We note that $\tilde{s}$ and s' are fully synchronized at $t+1$.

Taking expectation over the eight (four without a Class 2 arrival and four with a Class 2 arrival at $t-1$) scenarios together, we have

$$\begin{aligned}
&V_{t-1}^{\pi^*}(x) - V_{t-1}^{\pi'}(x) \\
=&\mathbb{E}[C_{t-1}(x, \pi^*(t-1)) + V_t^{\pi^*}(S_{t-1}(x, \pi^*(t-1))) \\
&- C_{t-1}(x, \pi'(t-1)) - V_t^{\pi'}(S_{t-1}(x, \pi'(t-1)))|A_2(t-1) = 1]\mathbb{P}(A_2(t-1) = 1)
\end{aligned}$$

Figure 16 Coupling illustration for Case 2b with one Class 2 customer and a Class 2 arrival at $t-1$.



$$
\begin{aligned}
&+\mathbb{E}[C_{t-1}(x,\pi^*(t-1))+V_t^{\pi^*}(S_{t-1}(x,\pi^*(t-1)))\\
&\quad -C_{t-1}(x,\pi'(t-1))-V_t^{\pi'}(S_{t-1}(x,\pi'(t-1)))|A_2(t-1)=0]\mathbb{P}(A_2(t-1)=0)\\
\geq&\mathbb{E}[C_{t-1}(x,\pi^*(t-1))+G_t(S_{t-1}(x,\pi^*(t-1)))\\
&\quad -C_{t-1}(x,\pi'(t-1))-V_t^{\pi'}(S_{t-1}(x,\pi'(t-1)))|A_2(t-1)=1]\mathbb{P}(A_2(t-1)=1)\\
&+\mathbb{E}[C_{t-1}(x,\pi^*(t-1))+V_t^{\pi^*}(S_{t-1}(x,\pi^*(t-1)))\\
&\quad -C_{t-1}(x,\pi'(t-1))-V_t^{\pi'}(S_{t-1}(x,\pi'(t-1)))|A_2(t-1)=0]\mathbb{P}(A_2(t-1)=0)\\
=&\mu_1c_1-\mu_2c_2>0,
\end{aligned}
$$

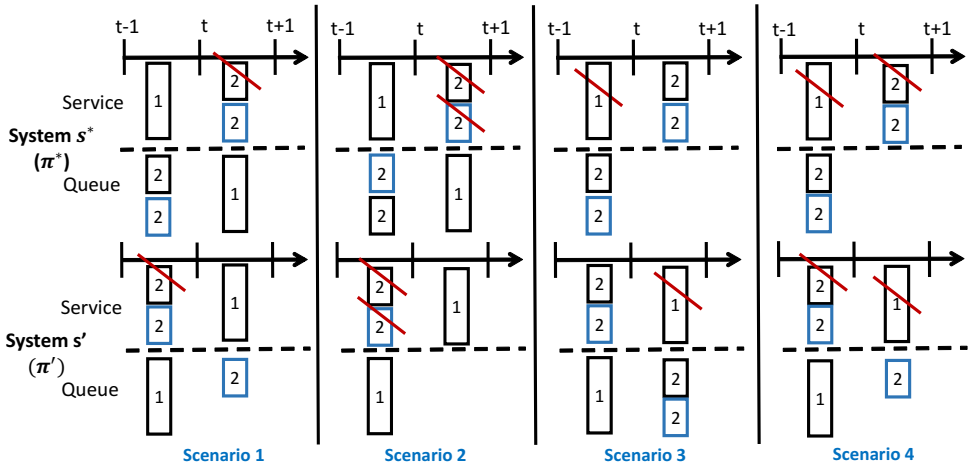
where the last inequality follows from the condition of Case 2b. This contradicts the assumption that π^* is the optimal.

- **If $x_1 \geq 1$ and $x_2 \geq 2$,** for π^* to deviate from $\hat{\pi}$, under π^*, we admit one Class 1 customer at $t-1$. Under π', we admit two Class 2 customers at $t-1$. At t, π' admits one Class 1 customer, and from $t+1$ onward, π' will follow $\hat{\pi}$. Similar to Case 1, there are 6 outcomes across the two systems at $t-1$. Their corresponding probabilities are also the same as in Case 1. From Figure 17, it is easy to see that the two systems are fully synchronized at $t+1$. Thus,

$$V_t^{\pi^*}(X)-V_t^{\pi'}(X)=2c_2\mu_2-c_1\mu_1>0,$$

where the last inequality follows from the condition of Case 2b. This contradicts the assumption that π^* is the optimal policy. $\square$

Figure 17 Coupling illustration for Case 2b with $X_2(t-1) \geq 2$.



A.3. An auxiliary lemma for the proof of Case 2b

We consider the benefits of adding an extra server in a single time slot. Given state x, let $z^G(x)$ denote a feasible server allocation strategy at this time slot, i.e.,

$$\sum_{i=1}^{I} z_i^G(x) \leq N+1, \quad \frac{z_i^G(x)}{m_i} \in \mathbb{N}_0, \quad \text{and} \quad 0 \leq \frac{z_i^G(x)}{m_i} \leq x_i.$$

Let $G_t(x)$ denote the optimal cost to go at time t with $X(t)=x$, when there are $N+1$ servers at t, and N servers from $t+1$ onwards.

LEMMA 3. *If an additional server is added at time epoch t for one time slot,*

$$G_t(x) \leq V_t^*(x).$$

Proof: We consider two coupled systems, s^* and $\tilde{s}$, that start from the same state at time t, i.e., $X(t)=x$, and see the same arriving customers. System s^* follows the optimal scheduling policy. System $\tilde{s}$ is identical to s^*, except that at time t, an additional server is added for that time slot only. Consider a feasible policy for $\tilde{s}$, under which the extra server is not utilized. Then, the two systems would incur the same holding cost. Since $G_t(x)$ is minimized over all feasible policies, the result directly follows. □

Appendix B: Proof of the Competitive Ratio Bound in Section 3.2

B.1. An Auxiliary Lemma

Before we prove Theorem 2, we first provide an auxiliary lemma.

LEMMA 4. *When $c_1\mu_1 < c_2\mu_2$, for any $t=0,\ldots,T-1$, if $X_1(t) \geq 1$ and $X_2(t) \geq 2$, $\pi^*(t) \neq P_1$.*

Lemma 4 indicates that in this parameter regime it is never optimal to schedule one Class 1 customer over two Class 2 customers.

Proof of Lemma 4 The proof is based on backwards induction.

Base Case: $t = T-1$**.** Due to the linearity of (4) in $\pi(t)$, it is straightforward to see that as $c_1\mu_1/2 < c_2\mu_2$, we would prefer two Class 2 customers over one Class 1 customer.

Inductive step. Suppose the claim is true for all s, $t \le s \le T-1$, for some $1 \le t \le T-1$. We show by contradiction that at $t-1$, the claim is also true. Suppose $X(t-1) = x$. We can restrict our attention to states x with $x_1 \ge 1$ and $x_2 \ge 2$. Suppose by contradiction that at $t-1$, P_1 is optimal, i.e., we prefer admitting one Class 1 customer over two Class 2 customers at $t-1$.

Following the same lines of analysis as in Case 1 of Theorem 1, we can construct a coupled System s' which operates under a suboptimal policy π'. In particular, π' admits two Class 2 customers at $t-1$ and one Class 1 customer at t. From $t+1$ onward, π' follows π^*. Then we can show that

$$V_t^{\pi^*}(x) - V_t^{\pi'}(x) = 2c_2\mu_2 - c_1\mu_1 > 0.$$

This contradicts the assumption that π^* is the optimal policy. □

B.2. Proof of Theorem 2.

We start by using a coupling argument to establish a bound on the difference in the number of customers between a system that follows the optimal policy and a system that follows $\mathbf{P_2^I}$. Specifically, consider two coupled systems that see exactly the same customers in terms of their arrival and service times. The system that follows π^* is denoted as s^* and the system that follows $\mathbf{P_2^I}$ is denoted as $\tilde{s}$.

Let $N_i(t)$ denote the number of Class i arrivals by time t; by our coupling this is the number of Class i arrivals in each system. Let $U_i^{*,k}(t)$, $k \le N_i(t)$, denote the remaining service time of the k-th Class i arrival at time t in the s^* system. $U_i^{*,k}(t) = 0$ implies that the customer has already left the system. Similarly, we denote $\tilde{U}_i^k(t)$ as the remaining service time for the k-th Class i arrival in $\tilde{s}$ at time t. Let $X_i^*(t)$ denote the number of Class i customers present in s^* at time t and $\tilde{X}_i(t)$ denote the number of Class i customers in $\tilde{s}$.

We now describe how we couple the scheduling policies in both systems.

System s^* follows the optimal scheduling policy and serves customers within each class in the first come first served (FCFS) order. Note that from Lemma 4, if there are at least two Class 2 customers in s^*, s^* will prioritize the Class 2 customers. System $\tilde{s}$ follows policy $\mathbf{P_2^I}$ and serves customers within each class according to FCFS with the following exceptions:

- At time $t \in \{0, 1, \ldots, T-1\}$, if s^* admits two Class 2 customers, i and j, and there are at least two Class 2 customers in $\tilde{s}$, we consider the following two scenarios. 1) If the same Class 2 customers admitted in s^*, i and j, are still in the $\tilde{s}$, we admit them in $\tilde{s}$ as well. 2) If the i and/or j Class 2 customers have already left $\tilde{s}$, but there are Class 2 customers in $\tilde{s}$, who have been served

more times in s^* (i.e., there are Class 2 customers, m and n, with $U_2^{*,m} < \tilde{U}_2^m$ and $U_2^{*,n} < \tilde{U}_2^n$), we admit those customers. Note that other than the above two scenarios, we admit two Class 2 customers in $\tilde{s}$ according to FCFS.

- At time $t \in \{0, 1, \ldots, T-1\}$, if s^* admits a Class 1 customer and there are at least two Class 2 customers in $\tilde{s}$, we consider the following scenario. If there are Class 2 customers in $\tilde{s}$ who have been served more times in s^* (i.e., there are Class 2 customers, m and n, with $U_2^{*,m} < \tilde{U}_2^m$ and $U_2^{*,n} < \tilde{U}_2^n$), we admit those customers. Note that other than the above-mentioned scenario, we admit two Class 2 customers in $\tilde{s}$ according to FCFS.

Next, we prove the following statement, which we refer to as Statement $\mathcal{S}$:

1. $\tilde{U}_2^k(t) \le U_2^{*,k}(t)$ for all $k = 1, \ldots, N_2(t)$, except at most one k^*, for which $\tilde{U}_2^{k^*}(t) = U_2^{*,k^*}(t) + \kappa_2(t)$ for some $\kappa_2(t) \in (0, T]$. If there is no such k^*, we set $\kappa_2(t) = 0$.

2. $\tilde{U}_1^k(t) \le U_1^{*,k}(t)$ for all $k = 1, \ldots, N_2(t)$, and $\sum_{k=1}^{N_1(t)} U_1^{*,k}(t) = \sum_{k=1}^{N_1(t)} \tilde{U}_1^k(t) + \kappa_1(t)$ for some $\kappa_1(t) \ge \kappa_2(t)$.

That is, all Class 2 jobs have been served more times in the $\tilde{s}$ system than the s^* system, except for at most one Class 2 job, k^*, which has been served $\kappa_2(t)$ more times in the s^* system than the $\tilde{s}$ system. Additionally, all Class 1 jobs have been served more times in the $\tilde{s}$ system than the s^* system. The total amount of additional Class 1 service times is $\kappa_1(t)$, which is at least as large as $\kappa_2(t)$.

Under Statement $\mathcal{S}$, we have

$$\begin{aligned} \tilde{X}_2(t) &= \sum_{k=1}^{N_2(t)} 1_{\{\tilde{U}_2^k(t) > 0\}} \le X_2^*(t) + 1_{\{\kappa_2(t) > 0\}}, \\ \tilde{X}_1(t) &= \sum_{k=1}^{N_1(t)} 1_{\{\tilde{U}_1^k(t) > 0\}} \le X_1^*(t). \end{aligned} \tag{14}$$

We next prove Statement $\mathcal{S}$ by induction following the coupled policies:

Base Case: At $t = 0$, the two systems starts from the same state. Thus, Statement $\mathcal{S}$ holds trivially.

Inductive step: Suppose Statement $\mathcal{S}$ is true at time t. We next show that it holds at time $t+1$ as well.

We divide the analysis into different cases depending on the value of $X^*(t)$ and $\tilde{X}(t)$.

Case I $X_2^*(t) \ge 2$. In this case, s^* admits two Class 2 customers at t.

Case Ia. If $\tilde{X}_2(t) \ge 2$, $\tilde{s}$ admits two Class 2 customers as well. $\kappa_1(t+1) = \kappa_1(t)$. If the admitted Class 2 customers are the same in the two systems, $\kappa_2(t+1) = \kappa_2(t)$. If the admitted Class 2 customers are not the same in the two systems and $\kappa_2(t) > 0$, customer k^* in Class 2 will be admitted in $\tilde{s}$ and $\kappa_2(t+1) = \kappa_2(t) - 1$. Otherwise, $\kappa_2(t) = 0$, which implies $\kappa_2(t+1) = 0$.

Case Ib. If $\tilde{X}_1(t) \geq 1$ and $\tilde{X}_2(t) \leq 1$, $\tilde{s}$ admits a Class 1 customer. $\kappa_1(t+1) = \kappa_1(t) + 1$. If i) $\tilde{X}_2(t) = 1$, ii) one of the two Class 2 customers admitted in s^* is the Class 2 customer remaining in $\tilde{s}$, and iii) that Class 2 customer has less of equal remaining service time in s^* than in $\tilde{s}$, $\kappa_2(t+1) = \kappa_2(t) + 1$. Otherwise, $\kappa_2(t+1) = \kappa_2(t) = 0$.

Case Ic. If $\tilde{X}_1(t) = \tilde{X}_2(t) = 0$, no customer is served in System $\tilde{s}$, $\kappa_1(t+1) = \kappa_1(t)$. $\tilde{X}_2(t) = 0$ implies that $\kappa_2(t) = 0$. Thus, $\kappa_2(t+1) = 0$.

Case II $X_1^*(t) \geq 1$, $X_2^*(t) = 1$, and $\pi^*(t) = P_2$. In this case, by (14), $\tilde{X}_2(t) \leq 2$.

Case IIa. If $\tilde{X}_2(t) = 2$, $\tilde{s}$ admits two Class 2 customers. $\kappa_1(t+1) = \kappa_1(t)$. As $\tilde{X}_2(t) > X_2^*(t)$, one of the two Class 2 customers admitted in $\tilde{s}$ is behind the corresponding Class 2 customer in s^*. Thus, $\kappa_2(t+1) = \kappa_2(t) - 1$.

Case IIb. If $\tilde{X}_1(t) \geq 1$ and $\tilde{X}_2(t) \leq 1$, $\tilde{s}$ admits a Class 1 customer. $\kappa_1(t+1) = \kappa_1(t) + 1$. If $\tilde{X}_2(t) = 1$, denote the Class 2 customer in $\tilde{s}$ as customer i. If $\tilde{U}_2^i(t) \geq U_2^{*,i}(t)$ (customer i could be the remaining Class 2 customer in s^* or could have already left s^*), $\kappa_2(t+1) = \kappa_2(t) + 1$. Otherwise, $\kappa_2(t+1) = \kappa_2(t) = 0$.

Case IIc. If $\tilde{X}_1(t) = 0$ and $\tilde{X}_2(t) = 1$, $\tilde{s}$ admits the Class 2 customer, which we denote as customer i. $\kappa_1(t+1) = \kappa_1(t)$. If customer i is still in s^*, $\kappa_2(t+1) = \kappa_2(t)$. If customer i has already left s^*, $\kappa_2(t) > 0$ and $\kappa_2(t+1) = \kappa_2(t) - 1$.

Case IId. If $\tilde{X}_1(t) = \tilde{X}_2(t) = 0$, no customer is served in $\tilde{s}$. $\kappa_1(t+1) = \kappa_1(t)$. $\tilde{X}_2(t) = 0$ implies that $\kappa_2(t) = 0$. Thus, $\kappa_2(t+1) = 0$.

Case III $X_1^*(t) \geq 1$, $X_2^*(t) = 1$, and $\pi^*(t) = P_2^I$, so that the s^* system admits a Class 1 customer. In this case, by (14), $\tilde{X}_2(t) \leq 2$.

Case IIIa. If $\tilde{X}_2(t) = 2$, $\tilde{s}$ admits two Class 2 customers. $\kappa_1(t+1) = \kappa_1(t) - 1$. As there is one more Class 2 customer in $\tilde{s}$ than in s^*, $\kappa_2(t) > 0$. Then, $\kappa_2(t+1) = \kappa_2(t) - 1$.

Case IIIb. If $\tilde{X}_1(t) \geq 1$ and $\tilde{X}_2(t) \leq 1$, $\tilde{s}$ admits a Class 1 customer as well. In this case, $\kappa_i(t+1) = \kappa_i(t)$ for $i = 1, 2$.

Case IIIc. If $\tilde{X}_1(t) = 0$ and $\tilde{X}_2(t) = 1$, $\tilde{s}$ admits the Class 2 customer. Since a Class 1 customer is admitted in s^*, $\kappa_1(t) > 0$ and $\kappa_1(t+1) = \kappa_1(t) - 1$. If $\kappa_2(t) > 0$, $\kappa_2(t+1) = \kappa_2(t) - 1$. Otherwise, $\kappa_2(t+1) = \kappa_2(t) = 0$.

Case IIId. If $\tilde{X}_1(t) = 0$ and $\tilde{X}_2(t) = 0$, $\kappa_1(t+1) = \kappa_1(t) - 1$ and $\kappa_2(t+1) = \kappa_2(t) = 0$.

Case IV $X_1^*(t) = 0$ and $X_2^*(t) \leq 1$. Based on the induction argument, $\tilde{X}_1(t) = 0$ and $\kappa_1(t) = 0$. As $\kappa_2(t) \leq \kappa_1(t)$, we have $\kappa_2(t) = 0$ and $\tilde{X}_2(t) \leq X_2^*(t)$. Note that if $X_2^*(t) = \tilde{X}_2(t) = 1$, the Class 2 customer in $\tilde{s}$ has less or equal remaining service time than the Class 2 customer in s^*. Thus, $\kappa_i(t+1) = \kappa_i(t) = 0$ for $i = 1, 2$.

Case V $X_1^*(t) \geq 1$ and $X_2^*(t) = 0$. In this case, s^* admits a Class 1 customer and by (14), $\tilde{X}_2(t) \leq 1$.

Case Va. If $\tilde{X}_1(t) \geq 1$, $\tilde{s}$ admits a Class 1 customer as well. Then, $\kappa_i(t+1) = \kappa_i(t)$ for $i = 1, 2$.

Case Vb. If $\tilde{X}_1(t)=0$ and $\tilde{X}_2(t)=1$, $\tilde{s}$ admits the Class 2 customer. Since $X_2^*(t)=0$, $\kappa_2(t)>0$. Then, $\kappa_i(t+1)=\kappa_i(t)-1$ for $i=1,2$.

Case Vc. If $\tilde{X}_1(t)=0$ and $\tilde{X}_2(t)=0$, no customer is served in $\tilde{s}$. Since there is at least one Class 1 customer in s^*, $\kappa_1(t)>0$ and $\kappa_1(t+1)=\kappa_1(t)-1$. Since $\tilde{X}_2(t)=0$, $\kappa_2(t+1)=\kappa_2(t)=0$.

The above cases cover all possible scenarios. We have thus shown that $\mathcal{S}$ holds at $t+1$ as well.

Based on Statement $\mathcal{S}$ (and hence, (14)), the following inequality holds path-by-path, i.e., for $t=0,\ldots,T$:

$$c_1\tilde{X}_1(t)+c_2\tilde{X}_2(t)\le c_1X_1^*(t)+c_2X_2^*(t)+c_2\cdot 1_{\{\kappa_2(t)>0\}}$$

In addition, when $\kappa_2(t)>0$, $\kappa_1(t)\ge\kappa_2(t)>0$ (from the second part of Statement $\mathcal{S}$). Thus,

$$1_{\{\kappa_2(t)>0\}}\le X_1^*(t) \text{ for } t=0,\ldots,T.$$

Through stochastic dominance, we have

$$\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1\tilde{X}_1(s)+c_2\tilde{X}_2(s)\right]\le\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)+c_21_{\{\kappa_2(s)>0\}}\right] \tag{15}$$

and

$$\mathbb{E}\left[\sum_{s=t+1}^{T-1} 1_{\{\kappa_2(s)>0\}}\right]\le\mathbb{E}\left[\sum_{s=t+1}^{T-1} X_1^*(s)\right]. \tag{16}$$

Then,

$$\begin{aligned}
\frac{V_t^{\mathbf{P_2^I}}(x)}{V_t^{\pi^*}(x)}=&\frac{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1\tilde{X}_1(s)+c_2\tilde{X}_2(s)\right]+\mathbb{E}\left[F_1(\tilde{X}_1(T))+F_2(\tilde{X}_2(T))\right]}{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)\right]+\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T))\right]}\\
\le&\frac{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)+c_21_{\{\kappa_2(s)>0\}}\right]}{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)\right]+\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T))\right]}\\
&+\frac{\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T)+1_{\{\kappa_2(T)>0\}})\right]}{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)\right]+\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T))\right]} \quad \text{by (15)}\\
=&1+\frac{\mathbb{E}\left[c_2\sum_{s=t+1}^{T-1} 1_{\{\kappa_2(s)>0\}}\right]+\mathbb{E}\left[F_2(1_{\{\kappa_2(T)>0\}})\right]}{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)\right]+\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T))\right]}\\
\le&1+\frac{\mathbb{E}\left[c_2\sum_{s=t+1}^{T-1} X_1^*(s)\right]+\mathbb{E}\left[F_2(X_1^*(T))\right]}{\mathbb{E}\left[\sum_{s=t+1}^{T-1} c_1X_1^*(s)+c_2X_2^*(s)\right]+\mathbb{E}\left[F_1(X_1^*(T))+F_2(X_2^*(T))\right]} \quad \text{by (16)}\\
\le&2 \text{ as } c_1\ge c_2.
\end{aligned}$$

□

Appendix C: Proofs of the Results in Section 5

We denote $\bar{\pi}^*$ as the optimal scheduling policy. We also denote $\bar{\pi}^{\mathrm{LP}}$ as the optimal policy induced by the LP (9), which we also refer to as the fluid $c\mu/m$ rule.

C.1. Proof of Lemma 1

We prove process level convergence by induction on t.

We first note that by assumption, $\bar{X}^\eta(0) \Rightarrow \bar{x}(0)$.

Suppose $\bar{X}^\eta(t) \Rightarrow \bar{x}(t)$ as $\eta \to \infty$. Then, as $\bar{\psi}^\eta \to \bar{\psi}$,

$$\bar{D}_i^\eta(t)|\bar{X}^\eta(t) \stackrel{D}{=} \frac{1}{\eta}\text{Binomial}(\eta\bar{\psi}_i^\eta(\bar{X}^\eta(t))/m_i, \mu_i) \Rightarrow \mu_i\bar{\psi}_i(\bar{x}(t))/m_i \text{ as } \eta \to \infty.$$

In addition, because

$$\bar{A}_i^\eta(t) \Rightarrow \lambda_i(t) \text{ as } \eta \to \infty$$

and $\bar{X}_i^\eta(t+1) = \bar{X}_i^\eta(t) + \bar{A}_i^\eta(t) - D_i^\eta(t)$, we have $\bar{X}_i^\eta(t+1) \Rightarrow \bar{x}_i(t+1)$ as $\eta \to \infty$.

C.2. Proof of Lemma 2

For ease of exposition, we sort the I customer classes in the decreasing order of their corresponding $c\mu/m$ index. Let $[i]$ denote the i-th class in this order.

The proof is based on backwards induction and an interchange argument. We first introduce $\bar{V}_t^{\bar{\pi}}(x)$ as the cost to go function under policy $\bar{\pi}$ for the fluid model, i.e.,

$$\bar{V}_t^{\bar{\pi}}(x) = \sum_{s=t+1}^{T-1}\sum_{i=1}^{I} c_i\bar{x}_i(s) + \sum_{i=1}^{I}\bar{F}_i(\bar{x}_i(T))$$

with $\bar{x}_i(t) = x_i$, and for $s = t+1, \ldots, T$,

$$\bar{x}_i(s) = \bar{x}_i(s-1) + \lambda_i(s-1) - \mu_i\bar{\pi}_i(s-1).$$

We define

$$\begin{aligned} V_{T-1}^{\bar{\pi}}(x) &= \sum_{i=1}^{I}\bar{F}_i(\bar{x}_i(T)) \quad \text{given } \bar{x}_i(T-1) = x \\ &= \sum_{i=1}^{I}\xi c_i(x_i + \lambda_i(T-1) - \mu_i\bar{\pi}_i(T-1)). \end{aligned}$$

Base Case: $t = T-1$**.** Let $z = (z_1, \ldots, z_I) \in \mathbb{R}_0^I$ denote the amount of service capacity allocated to each class. Then,

$$\begin{aligned} \min_{\bar{\pi}} V_{T-1}^{\bar{\pi}}(x) = \min_{z} \; & \sum_{i=1}^{I}\xi c_i(x_i + \lambda_i(T-1) - \mu_i z_i/m_i) \\ \text{s.t.} & \sum_{i=1}^{I} z_i \le N, \quad 0 \le z_i \le m_i x_i \text{ for } i = 1, \ldots, I. \end{aligned}$$

It is straightforward to see that the optimal solution is to prioritize according to the $c\mu/m$ index.

Inductive Step. Suppose it is optimal to follow the $c\mu/m$ rule for $s = t, \ldots, T-1$ and $0 \le t \le T-1$. We now consider the time epoch $t-1$. Let $\bar{x}(t-1) = x$ for some initial state x. Suppose by

contradiction that it is optimal to deviate from the $c\mu/m$ rule at time $t-1$. Then for some x, there exists i and j, having $[j]<[i]$, such $\bar{\pi}^*_j(t-1)>0$ while $x_i-\bar{\pi}^*_i(t-1)>0$, i.e., some service capacity is allocated to Class $[j]$ while there is still Class $[i]$ fluid waiting. If Class $[i]$ is served some time after $t-1$ under $\bar{\pi}^*$, let $\sigma\geq t$ be the first time after time $t-1$ at which Class $[i]$ is served under $\bar{\pi}^*$. Let $\epsilon:=\min\{\bar{\pi}^*_i(\sigma), x_i-\bar{\pi}^*_i(t-1), \bar{\pi}^*_j(t-1)m_j/m_i\}$. Consider a policy $\bar{\pi}'$ which is identical to $\bar{\pi}^*$, except that at time $t-1$, $\bar{\pi}'_i(t-1)=\bar{\pi}^*_i(t-1)+\epsilon$, $\bar{\pi}'_j(t-1)=\bar{\pi}^*_j(t-1)-\epsilon$, and at time σ, $\bar{\pi}'_i(\sigma)=\bar{\pi}^*_i(\sigma)-\epsilon$, $\bar{\pi}'_j(\sigma)=\bar{\pi}^*_j(\sigma)+\epsilon$, i.e., the ϵ capacity is swapped under $\bar{\pi}'$. Then,

$$\bar{V}^{\bar{\pi}^*}_{t-1}(x)-\bar{V}^{\bar{\pi}'}_{t-1}(x)=\epsilon(\sigma-t+1)\frac{c_i\mu_i}{m_i}-\epsilon(\sigma-t+1)\frac{c_j\mu_j}{m_j}>0.$$

If Class $[i]$ is not served after $t-1$, let $\epsilon=\min\{x_i-\bar{\pi}^*_i(t-1), \bar{\pi}^*_j(t-1)m_j/m_i\}$. Consider a policy $\bar{\pi}'$ which is identical to $\bar{\pi}^*$, except that at time $t-1$, $\bar{\pi}'_i(t-1)=\bar{\pi}^*_i(t-1)+\epsilon$, $\bar{\pi}'_j(t-1)=\bar{\pi}^*_j(t-1)-\epsilon$. Then,

$$\bar{V}^{\bar{\pi}^*}_{t-1}(x)-\bar{V}^{\bar{\pi}'}_{t-1}(x)=\epsilon(T-t+\xi)\frac{c_i\mu_i}{m_i}-\epsilon(T-t+\xi)\frac{c_j\mu_j}{m_j}>0.$$

This contradicts the optimality of $\bar{\pi}^*$. Thus, it is optimal to follow the $c\mu/m$ rule at $t-1$ as well. □

C.3. Proof of Theorem 3

The proof is decomposed into two steps.

Step 1: Prove that the optimal fluid value function is a lower bound for the stochastic value function. Formally, we shall prove the following lemma.

LEMMA 5. *For any Markovian policies* π^η,

$$V_0^{\pi^\eta,\eta}(x)/\eta\geq\bar{V}^*_0(x/\eta).$$

Note that $\bar{V}^*_0(x)$ is continuous in x when the $c\mu/m$-index takes distinct values (note that the optimal fluid trajectory is continuous in its initial condition). Thus, if $x^\eta/\eta\to x$ as $\eta\to\infty$,

$$\liminf\eta\to\infty V_0^{\pi^\eta,\eta}(x^\eta)/\eta\geq\liminf\eta\to\infty\bar{V}^*_0(x^\eta/\eta)=\bar{V}^*_0(x).$$

Step 2: Prove that the idle-aware $c\mu/m$ policy achieves the lower bound asymptotically. Formally, we first have the following lemma. Let $\psi^{\mathrm{IP}(\Gamma),\eta}$ denote the mapping from the state of the system to the allocation of servers according to the idle-aware $c\mu/m$ rule with parameter $\Gamma\geq 0$. We also write $\bar{\psi}^{\mathrm{LP}}$ as the mapping corresponding to the $c\mu/m$ rule for the fluid model.

LEMMA 6. *For any fixed* $\Gamma\geq 0$, $\bar{\psi}^{IP(\Gamma),\eta}\to\bar{\psi}^{LP}$, *as* $\eta\to\infty$.

Under the assumption that the $c\mu/m$ index takes distinct values, the LP (9) has a unique optimal solution:

$$\bar{\psi}^{\mathrm{LP}}_{[j]}(x)=\left(N-\sum_{i=1}^{j-1}\bar{\psi}^{\mathrm{IP}}_{[i]}(x)\right)\wedge x_{[j]}m_{[j]},$$

and the mapping $\bar{\psi}^{\mathrm{LP}}(x)$ is continuous in x. Thus, from Lemma 1, we have

$$\bar{X}^{\pi^{\mathrm{IP}(\Gamma),\eta},\eta} \Rightarrow \bar{x}^{\bar{\pi}^{\mathrm{LP}}} \text{ uniformly on } [0,T] \text{ as } \eta \to \infty. \tag{17}$$

We next establish the uniform integrability of $\bar{X}^{\pi^{IP(\Gamma),\eta},\eta}$. We shall drop the superscript $\pi^{IP(\Gamma),\eta}$ as it can be clearly understood from the context. Because $\sup_{0\le t\le T} \bar{X}_i^\eta(t) \le \sum_{t=1}^T \bar{A}_i^\eta(t)$, we have

$$\begin{aligned}
\sup_\eta \mathbb{E}\left[\left(\sup_{0\le t\le T} \bar{X}_i^\eta(t)\right)^2\right] &\le \sup_\eta \mathbb{E}\left[\left(\sum_{t=1}^T \bar{A}_i^\eta(t)\right)^2\right] = \sup_\eta \frac{1}{\eta^2}\mathbb{E}\left[\left(\sum_{t=1}^T A_i^\eta(t)\right)^2\right] \\
&= \sup_\eta \frac{1}{\eta^2}\left[\mathrm{Var}\left(\sum_{i=1}^T A_i^\eta(t)\right) + \left[\mathbb{E}\left(\sum_{t=1}^T A_i^\eta(t)\right)\right]^2\right] \\
&= \sup_\eta \frac{1}{\eta^2}\left[\sum_{i=1}^T \mathrm{Var}\left(A_i^\eta(t)\right) + \left[\sum_{t=1}^T \mathbb{E}\left(A_i^\eta(t)\right)\right]^2\right] \\
&= \sup_\eta \frac{1}{\eta}\left[\sum_{i=1}^T \lambda_i(t)\left(1-\lambda_i(t)\right)\right] + \left[\sum_{t=1}^T \lambda_i(t)\right]^2 < \infty.
\end{aligned}$$

This implies the uniform integrability of $\bar{X}^{\pi^{\mathrm{IP}(\Gamma),\eta},\eta}$, which combining with (17) indicates that if $x^\eta/\eta = x$ as $\eta \to \infty$, then

$$\lim_{\eta\to\infty} V_0^{\pi^{\mathrm{IP}(\Gamma),\eta},\eta}(x^\eta)/\eta = \bar{V}_0^{\bar{\pi}^{\mathrm{LP}}}(x).$$

Lastly, by Lemma 2, we have $\bar{V}_0^{\bar{\pi}^{LP}}(x) = \bar{V}_0^*(x)$.

C.3.1. Proof of Lemma 5 We first note that for any feasible Markovian policy π^η, we have

$$\frac{1}{\eta} V_0^{\pi^\eta,\eta}(x) = \sum_{t=1}^{T-1}\sum_{i=1}^{I} \mathbb{E}[c_i \bar{X}_i^\eta(t)] + \sum_{i=1}^{I} \mathbb{E}\left[F_i\left(\bar{X}_i^\eta(T)\right)\right]$$

and

$$\begin{aligned}
\mathbb{E}[\bar{X}_i^\eta(t+1)] &= \mathbb{E}[\bar{X}_i^\eta(t)] + \mathbb{E}[\bar{A}_i^\eta(t)] - \mathbb{E}[\bar{D}_i^\eta(t)] \\
&= \mathbb{E}[\bar{X}_i^\eta(t)] + \lambda_i(t) - \mu_i \mathbb{E}[\bar{\pi}_i^\eta(t-1)],
\end{aligned}$$

where $\bar{\pi}_i^\eta(t) = \pi_i^\eta(t)/\eta$ and satisfies that $\sum_{i=1}^I m_i \bar{\pi}_i^\eta(t-1) \le N$ and $0 \le \bar{\pi}_i^\eta(t-1) \le \bar{X}_i^\eta(t-1)$. This further implies that $\sum_{i=1}^I m_i \mathbb{E}[\bar{\pi}_i^\eta(t-1)] \le N$ and $0 \le \mathbb{E}[\bar{\pi}_i^\eta(t-1)] \le \mathbb{E}[\bar{X}_i^\eta(t-1)]$.

Next, if we set $\bar{x}_i(t) = \mathbb{E}[X_i^\eta(t)]/\eta$ and $\bar{\pi}_i(t) = \mathbb{E}[\pi_i^\eta(t)]/\eta$, then $(\bar{x}, \bar{\pi})$ constitutes a feasible solution to the fluid optimization problem (8). Thus, $V_0^{\pi^\eta,\eta}(x)/\eta \ge \bar{V}_0^*(x/\eta)$.

C.3.2. Proof of Lemma 6 For a given x^η, we denote $z^\eta = \bar{\psi}^{\mathrm{IP}(\Gamma),\eta}(x^\eta/\eta)$, i.e., it is the fluid-scaled optimal solution to the IP (6). Suppose $x^\eta/\eta \to x$ as $\eta \to \infty$. Let $\bar{z} = \bar{\psi}^{\mathrm{LP}}(x)$, i.e., it is the optimal solution to the LP (9). We also write $\tilde{z}^\eta = \lfloor \eta \bar{z} \rfloor/\eta$. Note that $\eta\tilde{z}^\eta$ is a feasible solution to the IP (6). This implies that $R(\eta\tilde{z}^\eta) \le R(\eta z^\eta)$. Then, for any $\eta \ge 1$, we have

$$\bar{R}(\bar{z}) \ge \bar{R}(z^\eta) \ge \bar{R}(\tilde{z}^\eta).$$

Next,

$$0 \le \bar{R}(\bar{z}) - \bar{R}(z^\eta) \le \bar{R}(\bar{z}) - \bar{R}(\tilde{z}^\eta) \le \sum_{i=1}^{I} \left(\frac{c_i \mu_i}{m_i} + \Gamma \right) \frac{1}{\eta} \to 0 \text{ as } \eta \to \infty.$$

Thus, $\bar{R}(z^\eta) \to \bar{R}(\bar{z})$ as $\eta \to \infty$.

Lastly, as the LP (9) has a unique optimal solution when the $c\mu/m$ index takes distinct values, we have $z^\eta \to \bar{z}$ as $\eta \to \infty$.

Appendix D: Proof of Theorem 4

The proof of Theorem 4 follows a similar sample-path construction as that in Armony and Bambos (2003). Fix a sample path ω, suppose by contradiction

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(t) = \delta > 0. \tag{18}$$

Then, there exists an increasing unbounded sequence of times $\{t_u\}_{u=1}^{\infty}$ such that

$$\lim_{u\to\infty} \frac{1}{t_u} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(t_u) = \delta.$$

We partition the state space into two mutually disjoint sets, such that for any $\psi \in \Omega_m$:

Set A includes all states for which there are some idle servers under ψ. Under Assumption 2, there are no jobs waiting under ψ for the states in set A .

Set B includes all states for which there is no idle server under ψ.

Define $\hat{s}_u = \sup\{t < t_u, X(t) \in A\}$, i.e., $\hat{s}_u$ is the last time before t_u at which $X(t) \in A$. By convention, we set $\hat{s}_u = 0$ if $X(t)$ has always been outside A. Then, $\liminf_{u\to\infty}(t_u - \hat{s}_u)/t_u = \epsilon_1$, for some $\epsilon_1 \in (0,1]$.

This implies we can find a further increasing unbounded subsequence $\{t_v\}_{v=1}^{\infty}$ such that $\lim_{v\to\infty}(t_v - \hat{s}_v)/t_v = \epsilon_1$. (With a little abuse of notation, we index v from 1 to ∞ as well.) We next construct an increasing unbounded sequence of times $\{s_v\}_{v=1}^{\infty}$ based on $\hat{s}_v$. Fix $\epsilon_2 \in (0,1)$ and set $s_v = \max\{\hat{s}_v, (1-\epsilon_2)t_v\}$. Our construction of s_v implies that

$$\lim_{v\to\infty} \frac{t_v - s_v}{t_v} = \epsilon_3 = \min\{\epsilon_1, \epsilon_2\} \in (0,1).$$

Note that $X(t) \notin A$ throughout the interval $(s_v, t_v]$ and

$$X_i(t_v) - X_i(s_v) = \sum_{t=s_v+1}^{t_v} A_i(t) - \sum_{t=s_v+1}^{t_v} D_i(t) = \sum_{t=s_v+1}^{t_v} A_i(t) - \sum_{k=1}^{K} T_k(s_v+1, t_v) \frac{\mu_i \psi_i^k}{m_i},$$

where $T_k(s_v+1, t_v)$ denotes the amount of time the system spends using the service configuration k during the interval $[s_v+1, t_v]$. Note that here we only use feasible configurations for a given system state. Thus, $\sum_{k=1}^{K} T_k(s_v+1, t_v) = t_v - s_v - 1$. As $t_v - s_v \to \infty$ as $v \to \infty$,

$$\liminf_{v\to\infty} \frac{\sum_{i=1}^{I} \frac{m_i}{\mu_i} (X_i(t_v) - X_i(s_v))}{t_v - s_v - 1} = \sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{\mu_i} - \sum_{k=1}^{K} \gamma_k \sum_{i=1}^{I} \phi_i^k, \tag{19}$$

where $\gamma_k \geq 0$, $k = 1, \ldots, K$, and $\sum_{k=1}^{K} \gamma_k = 1$, i.e., γ_k is the proportion of time configuration ϕ^k is used. Next, we divide the K service configurations into two groups: Φ^I which includes the configurations that do not utilize all servers, and Φ^{NI} which includes the configurations that utilize all servers. When $X(t) \notin A$, all the configurations utilized (i.e., with $\gamma_k > 0$) under Ω_M are in ϕ^{NI}. Then, we can rewrite the limit in (19) as

$$\sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{\mu_i} - N \sum_{k \in \Phi^{NI}} \gamma_k = \sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{\mu_i} - N \tag{20}$$

Next, we prove that the expression in (20) is less than or equal to 0. Suppose by contradiction that

$$\sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{\mu_i} > N \tag{21}$$

Since $\bar{\lambda} \in \mathcal{M}$,

$$\sum_{i=1}^{I} \frac{\bar{\lambda}_i m_i}{\mu_i} \leq N \sum_{k \in \Phi^{NI}} \alpha_k + \sum_{k \in \Phi^I} \alpha_k \sum_{i=1}^{I} \phi_i^k, \tag{22}$$

for some $\alpha_k \geq 0$, $k = 1, \ldots, K$, and $\sum_{k=1}^{K} \alpha_k = 1$. Combining (21) and (22) yields that

$$N < N \sum_{k \in \Phi^{NI}} \alpha_k + \sum_{k \in \Phi^I} \alpha_k \sum_{i=1}^{I} \phi_i^k < N,$$

where the last inequality follows because $\sum_{i=1}^{I} \phi_i^k < N$ for $k \in \Phi^I$. We get a contradiction. Thus,

$$\liminf_{v \to \infty} \frac{\sum_{i=1}^{I} \frac{m_i}{\mu_i} \left(X_i(t_v) - X_i(s_v) \right)}{t_v - s_v - 1} \leq 0.$$

This further implies that

$$\begin{aligned}
\liminf_{v \to \infty} \frac{1}{s_v} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(s_v) &= \liminf_{v \to \infty} \frac{1}{t_v(1-\epsilon_3)} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(s_v) \\
&\geq \liminf_{v \to \infty} \frac{1}{t_v(1-\epsilon_3)} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(t_v) \\
&= \frac{\delta}{1-\epsilon_3} > \delta,
\end{aligned}$$

which contradicts the assumption in (18). Thus,

$$\lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{I} \frac{m_i}{\mu_i} X_i(t) = 0, \tag{23}$$

and note that the argument here applies sample-path wise. The convergence in (23) further implies that the system is rate stable by Lemma 2.2 in Armony and Bambos (2003). $\square$