Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Scheduling Homogeneous Impatient Customers

Achal Bassamboo, Ramandeep Singh Randhawa

To cite this article:
Achal Bassamboo, Ramandeep Singh Randhawa (2016) Scheduling Homogeneous Impatient Customers. Management Science 62(7):2129-2147. http://dx.doi.org/10.1287/mnsc.2015.2241

MANAGEMENT SCIENCE
Vol. 62, No. 7, July 2016, pp. 2129–2147
ISSN 0025-1909 (print) | ISSN 1526-5501 (online)

http://dx.doi.org/10.1287/mnsc.2015.2241

Scheduling Homogeneous Impatient Customers

Achal Bassamboo
Kellogg School of Management, Northwestern University, Evanston, Illinois 60208,
a-bassamboo@northwestern.edu

Ramandeep Singh Randhawa
Marshall School of Business, University of Southern California, Los Angeles, California 90089,
ramandeep.randhawa@marshall.usc.edu

Customer impatience has become an integral component of analyzing services, especially in the context of call centers. Typically, when customers arrive to such systems, they seem identical or homogeneous; however, from the system's perspective, as they wait in the queue, their residual willingness to wait changes. For instance, a customer who has already waited for 10 minutes may have a different residual willingness to wait compared with a customer who has only waited for 1 minute. In this manner, as time progresses, customers become differentiated on their estimated patience levels. We exploit this dimension of customer heterogeneity to construct scheduling policies in overloaded systems that dynamically prioritize customers based on their time in queue to optimize any given system performance metric. Interestingly, the optimal policy has a very simple structure, and we find that implementing it can lead to significant improvements over the first-come, first-served policy.

Keywords: queues; applications; optimization; priority; approximations
History: Received August 27, 2013; accepted April 17, 2015, by Noah Gans, stochastic models and simulation. Published online in *Articles in Advance* July 27, 2015.

1. Introduction

Customers waiting for a service are typically impatient and may leave if the wait is long. Such impatience has been modeled extensively in the call center literature,[1] and a commonly used model is that customers are endowed with a "patience clock" and that they abandon when this clock runs out while they are still waiting for service. Typically, customer patience clocks are modeled by assuming that these are independent and identically distributed draws from a common distribution, referred to as the patience or abandonment distribution. In this paper, we use the fact that when a customer waits in the queue for a certain amount of time without abandoning, this reveals additional information to the system manager about their residual patience time. So, even though all customers may appear identical when they first join the queue, as time progresses, they become differentiated on their residual patience, and the system may benefit by incorporating this differentiation into scheduling decisions. This is precisely the main research question in this paper—What is the optimal way to schedule customers given that as customers wait in the queue, they reveal additional information about their individual patience levels?

Formally, we consider a queueing system with a stream of customers arriving to a single pool of identical servers that processes these customers. Each customer is associated with a service time, which reflects his or her service requirement, and a patience time, which is the maximum time the customer is willing to wait in the queue before his or her service commences. We assume that the service and patience times may be correlated for a given customer, but are independent and identically distributed across customers, and the joint distribution is known to the system manager. The correlation between service and patience times seems reasonable and has been observed empirically in call center applications in Reich (2012) and Mandelbaum and Zeltyn (2013). Our goal is to identify the optimal scheduling policy that differentiates between customers who have waited different amounts of time in the queue to minimize *any* given performance metric. For concreteness, we focus on three commonly used performance metrics in this paper, namely, the long-term abandonment rate of customers, the expected steady-state queue length,[2] and the expected offered wait.

[1] We refer the reader to Gans et al. (2003) and Aksin et al. (2007) for surveys of the call center literature.

[2] The expected steady-state queue length directly relates to the expected time experienced in the queue by all customers (who get processed or abandon) through Little's law. This waiting time metric is different from the offered wait metric, which focuses on the wait experienced by a hypothetical customer who does not abandon.

For tractability, we use a fluid-based approach that is commonly used for approximate analysis of large-volume systems. In particular, we assume that work arrives to the system as a fluid at a fixed rate and is also processed at a fixed rate. The fluid model simplifies our analysis considerably and yields interesting insights for the overloaded regime, in which the arrival rate exceeds the system capacity. We begin our analysis by proving that non–first-come, first-served (FCFS) scheduling policies can be implemented by creating multiple classes of customers, and each arriving customer is routed to one of these classes. Each such customer class has dedicated capacity allocated to it and operates in an FCFS manner. So, the optimal scheduling problem amounts to solving a class-based optimization problem by computing the optimal number of classes and the arrival rate and capacity allocated to each class. The fact that, within each class, customers are processed in an FCFS manner allows us to easily compute the performance metrics, and thus this class-based approach makes the problem more tractable. It turns out that the class-based optimization problem has an important structural property—the optimal number of classes is at most *two* for any set of parameters. Using this result, we characterize the structure of the optimal solution for any general performance metric and when service times and patience times are potentially correlated. Once the optimal class-based policy is identified, we provide a simple dynamic policy, which we refer to as a time-in-queue (TIQ) policy, that yields the same performance but prioritizes customers based on their time in the queue without splitting the customers into classes.

We obtain additional insights focusing on the case in which service and patience times are independent. Considering the queue length metric, it turns out that in this case the policy that minimizes the expected steady-state queue length depends intimately on the hazard rate of the patience times. If the hazard rate is monotone, then the optimal policies turn out to be either FCFS or last come, first served (LCFS). In particular, if the hazard rate of the patience times is monotone decreasing, then the optimal policy is FCFS. This result is intuitive because decreasing hazard rates imply that customers who have waited longer are less likely to abandon compared with those who have waited less. Thus, to minimize the queue length, the manager should focus on processing customers who are more likely to stay in the queue, and this is precisely what the FCFS policy does. Analogously, if the hazard rate of the patience times is monotone increasing, then the optimal policy is LCFS. Interestingly, under the fluid approach, the overall abandonment rate remains the same for all nonidling policies. Hence, the optimal policy improves the queue length without causing additional abandonments. For non-monotone hazard rates, the optimal policy that minimizes the expected queue length may take us beyond LCFS and FCFS, and we explicitly characterize the solution for quasi-concave hazard rates. This solution splits the customers into two classes, with one class being served with negligible waits and the other class being offered a wait that exceeds that under the FCFS policy. We also consider the objective of minimizing the expected steady-state offered wait of customers. In this case, it turns out that the customers' actual patience times are important, and correspondingly, the density function of the patience distribution plays the role that the hazard rate function played while minimizing expected steady-state queue length. For instance, if the density function is monotone decreasing, then we find that FCFS is optimal.

Finally, we tie our fluid analysis back to actual queueing systems. We use numerical experiments to compare the optimal TIQ policy with FCFS as a benchmark. We find that for the cases in which the TIQ policy differs from FCFS, implementing it can lead to significant performance improvement. Our work thus suggests that using updated estimates of customers' willingness to wait for scheduling decisions can improve performance, and that this can be done using policies that are fairly simple to implement.

2. Related Literature

Our work relates to the growing literature on optimal scheduling in queueing systems with reneging customers that utilizes customer heterogeneity. Unlike our work, this literature assumes that customers are a priori differentiated based on some characteristic, such as patience time distribution, service time distribution, or delay cost. A common mode of analysis in this literature is to use approximations to the original system derived under asymptotic regimes. (A notable exception is Down et al. 2011, which analyzes a two-class system with a single server.) Focusing on diffusion approximations, Dai and Tezcan (2008) develops robust control policies for a parallel server system to minimize the sum of holding and reneging costs when service times are pool dependent, Gurvich and Whitt (2010) proposes controls for parallel server systems that maintain fixed-queue ratios to meet service level constraints, and Ghamami and Ward (2013) considers a two-class, two-server N-model queueing system and proposes a threshold-based control policy to minimize system costs. Papers that utilize fluid approximations include Bassamboo et al. (2006), which proposes joint staffing and control for a parallel server system with time varying arrival rates, and Atar et al. (2010), which analyzes a multi-class single pool (V-model) system and proves that a

simple modification to the $c\mu$-priority rule is optimal on the fluid scale. (Atar et al. 2014 extends this result, which was proven under exponential service times to general service times.)

In contrast to the above literature, we assume that customers are a priori homogeneous, but become differentiated as they wait in the queue. This differentiation occurs because from the system's perspective, the remaining patience times of customers depends on the time they have already waited. Note that this differentiation occurs if patience times are nonexponentially distributed, and there is indeed empirical support for this (see, for instance, Mandelbaum and Zeltyn 2013).

Our analysis builds on the work in Whitt (2006) that proposes a fluid model to analyze a $G/GI/s+GI$ queueing system under the FCFS policy. (Kang and Ramanan 2010 and Zhang 2013 formally prove that this approximation is asymptotically accurate for large queueing systems.) In particular, we extend the analysis in Whitt (2006) to the case in which service and patience times are correlated (for evidence of such correlation in call centers, we refer the reader to Reich 2012 and Mandelbaum and Zeltyn 2013), and further because we can reduce our scheduling problem into a class-based analysis in which each class operates in an FCFS manner, the single-class fluid analysis directly applies. There is evidence that in the overloaded regime, such fluid approximations yield extremely accurate approximations to the underlying queueing system (see Bassamboo and Randhawa 2010, Bassamboo et al. 2010). Motivated by this, we directly work with the fluid model in this paper.

Our work also relates to the literature on due-date scheduling (see, for instance, Pinedo 2012). This literature typically looks at scheduling of jobs in manufacturing environments in which each job has an associated due date. In our system, impatient customers can be thought of as having their own deadlines; however, the key difference is that in our setting these due dates are binding, and a customer leaves the system as soon as his or her patience runs out. Furthermore, in many manufacturing settings, the manager knows the due dates of the jobs, whereas in our setting this information is not explicitly known. Nevertheless, the manager has distributional information, and as a customer waits in the queue, the manager can use this information to compute the conditional distribution of the remaining patience time. Notice that such an estimate of the patience time changes as the customer waits in the queue. In this paper, we show how this distributional estimate, which is essentially a mapping from time-in-queue to the conditional patience time, can be used to optimally schedule customers.

3. Model

In this paper we analyze $GI/GI/n+GI$ queueing systems using a fluid approach. We begin by describing the $GI/GI/n+GI$ system. We assume that customers arrive to this system as a renewal stream with rate Λ. Customers are impatient, and their work requirements may depend on their patience times. In particular, we use $f(x,y)$ to represent the density corresponding to the joint distribution of service time and the patience time associated with a customer. We denote the mean service requirement of an arriving customer by m, i.e.,

$$m=\int_0^\infty\int_0^\infty xf(x,y)\,dy\,dx.$$

We assume that f is strictly positive and m is finite. The system has n servers, each working at unit rate.

Our goal is to find the optimal scheduling policy to optimize the system performance. We will consider three performance metrics: overall abandonment rate, expected steady-state queue length, and the offered wait, which is defined as the expected steady-state time in queue that a customer with infinite patience would wait before entering service. Our approach will be to use a fluid version of this queueing system and find the optimal fluid-based scheduling policy. Then, we will use the fluid-based insights to propose a scheduling policy for $GI/GI/n+GI$ queueing systems (§5).

3.1. A Fluid Model

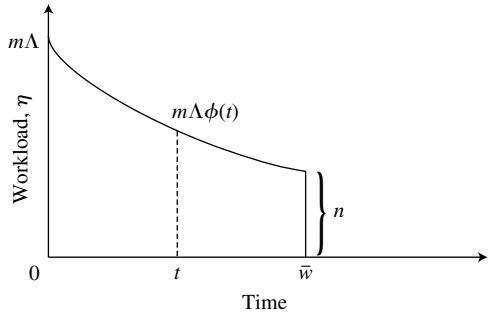
In the fluid model of the above $GI/GI/n+GI$ system, customers arrive in the form of a continuous flow at the constant rate Λ, and the system operates at a fixed rate n, i.e., n units of work can be processed per unit time. We assume that the system operates in the *overloaded regime* with the demand exceeding capacity, i.e., the system load $\rho:=(m\Lambda)/n>1$.

In this paper, because we are interested in steady-state performance measures, we need to characterize the steady state of the fluid system. For this, we define a nonincreasing function ϕ that captures the evolution of work to be processed in the system. In particular, for any arriving unit of customer work,

$$\phi(w):=\frac{1}{m}\int_{x=0}^\infty\int_{y=w}^\infty xf(x,y)\,dy\,dx$$

represents the fraction of its work that remains in the system after w time units of its arrival in the absence of any capacity allocation; that is, after w time units of arrival of one unit of customer work, $1-\phi(w)$ amount of work is depleted due to customer abandonments. In this sense, we refer to ϕ as the work depletion function. We will use the terminology workload at a time instant to refer to the total work in the system at that time, i.e., the total processing time of all current customers in the system.

Figure 1 Steady-State Workload Profile Under FCFS



3.1.1. Analyzing the FCFS Policy. If this system operates in an FCFS manner, then the steady-state behavior is characterized by a waiting time $\bar{w}$ so that all customers wait $\bar{w}$ time units before being processed. This waiting time $\bar{w}$ solves the following equation:

$$m\Lambda\phi(\bar{w}) = n. \tag{1}$$

This steady-state equation can be understood as follows. The term $m\Lambda$ in the left-hand side is the rate at which work is arriving to the system. So, the corresponding work that remains to be processed after customers have waited for $\bar{w}$ time units equals $(m\Lambda)\phi(\bar{w})$. In steady state, this must equal the rate at which work can be processed by the system, which is n. Note that because $m\Lambda > n$ and ϕ is continuous and decreasing with $\phi(0) = 1$ and $\lim_{w\to\infty}\phi(w) = 0$, (1) always has a unique solution. We would like to point out here that our model is a general version of the fluid model of the $G/GI/s+GI$ queue studied in Whitt (2006). In particular, in that model service times and patience times are independent, and we have $\phi(w) = m\bar{F}_A(w)$, where $\bar{F}_A(w)$ is the marginal tail cumulative distribution of patience times.

It is also useful to understand the steady-state workload profile under FCFS, which is illustrated in Figure 1. This workload profile will play an important role in the analysis of general policies, as we will see in the next section. As mentioned earlier, under the FCFS policy, in steady state, all customers wait exactly $\bar{w}$ time units before entering service. Work enters the system at rate $m\Lambda$, and after w time units of waiting in the queue, only $\phi(w)$ proportion of this work remains in the system. Thus, denoting the workload profile by η, we have $\eta(t) = m\Lambda\phi(t)$ for $t < \bar{w}$. Furthermore, because all work enters service after $\bar{w}$ time units, we have $\eta(t) = 0$ for $t \geq \bar{w}$.

3.1.2. Analyzing Non-FCFS Policies. A general scheduling policy in the fluid model can be thought of as a way of "applying" the capacity to the customers. In particular, FCFS can be viewed as applying the entire capacity at the waiting time of $\bar{w}$; i.e., customers wait for $\bar{w}$ time units and then get processed. Extending this notion to a general policy would entail applying the capacity at different waiting times. We focus on a class of policies in which each policy is described by a pair $(\kappa, v) \in \mathbb{R}_+^L \times \mathbb{R}_+^L$ for some finite $L \geq 1$, so that the policy applies capacity κ_i at the times $v_i \in \mathbb{R}_+$ for $1 \leq i \leq L$, where $v_j > v_i$ for $j > i$, and we have $\sum_{i=1}^L \kappa_i \leq n$. (We discuss the generality of this policy class in Remark 1 in §3.2.)

To characterize the steady state of these non-FCFS policies, we will adapt the methodology introduced in Whitt (2006) for FCFS systems. To illustrate this, let us consider a policy (κ, v) that applies capacities κ_1, κ_2 at times v_1, v_2, respectively, with $v_1 < v_2$. Figure 2(a) depicts the corresponding workload profile. Notice that for $t < v_1$, the workload evolves exactly as in the case of FCFS; i.e., we have $\eta(t) = m\Lambda\phi(t)$ for $t < v_1$. Furthermore, because the capacity κ_1 is applied at time v_1, the workload instantaneously drops at $t = v_1$ to $\eta(v_1) = m\Lambda\phi(v_1) - \kappa_1$. As time proceeds beyond v_1, the workload once again decays due to customer abandonments, and we have

$$\eta(t) = \eta(v_1)\frac{\phi(t)}{\phi(v_1)} = \left(m\Lambda - \frac{\kappa_1}{\phi(v_1)}\right)\phi(t), \quad \text{for } v_1 \leq t < v_2,$$

where $\phi(t)/\phi(v_1)$ denotes the conditional decay in the work of the remaining customers. At $t = v_2$, since we apply the capacity κ_2, the workload instantaneously drops to $\eta(v_2) = \eta(v_1)(\phi(v_2)/\phi(v_1)) - \kappa_2$. Finally, we have

$$\eta(t) = \eta(v_2)\frac{\phi(t)}{\phi(v_2)} = \left(m\Lambda - \frac{\kappa_1}{\phi(v_1)} - \frac{\kappa_2}{\phi(v_2)}\right)\phi(t), \quad \text{for } t > v_2.$$

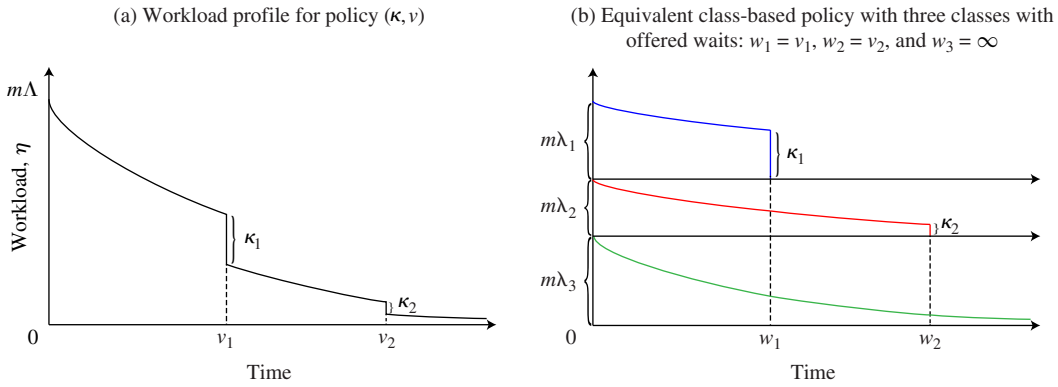
This argument easily extends to any general policy (κ, v), so that the corresponding workload profile is given by

$$\eta(t) = \left(m\Lambda - \sum_{\{j:\, v_j \leq t\}} \frac{\kappa_j}{\phi(v_j)}\right)\phi(t), \quad t \geq 0. \tag{2}$$

This representation also gives us the condition for feasibility of a given policy, which entails that the capacity allocated does not exceed the workload at any time instant, i.e., $\sum_{\{j:\, v_j \leq t\}}(\kappa_j/\phi(v_j)) \leq m\Lambda$ for all $t > 0$. Thus, we formally define the set of feasible policies as $(\mathcal{K}, \mathcal{V}) \subseteq \bigcup_{L\in\mathbb{N}} \mathbb{R}_+^L \times \mathbb{R}_+^L$ so that $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$ if $v_j > v_i$ for $j > i$, $\sum_{i=1}^{|\kappa|} \kappa_i \leq n$, and $\sum_{\{j:\, v_j \leq t\}}(\kappa_j/\phi(v_j)) \leq m\Lambda$ for all $t > 0$; we use the convention $|a|$ to denote the dimension of a vector a. In the rest of this paper, we will focus our attention on feasible policies.

An Equivalent View of the System. We next interpret any feasible policy (κ, v) with workload profile (2) in an alternative manner that will have an equivalent workload profile. To do so, we split customers into M classes at the time of arrival, with the arrival rate to

Figure 2 (Color online) Steady-State Workload Profile for a Policy That Applies Capacities κ_1, κ_2 at Times v_1, v_2, Respectively



class i denoted by λ_i, $i = 1, \ldots, M$, for some $M = L$ or $M = L + 1$ (this ambiguity will be resolved shortly). The arrival rate to class i, $i = 1, \ldots, L$, is chosen so that if this class were to operate in an FCFS manner with arrival rate λ_i and processing capacity κ_i, then it would have a steady-state offered wait of v_i. Thus, the arrival rate to class i, $i = 1, \ldots, L$, satisfies the relation

$$m\lambda_i \phi(v_i) = \kappa_i. \tag{3}$$

If $\sum_{i=1}^{L} \lambda_i = \Lambda$, then the L classes constructed above account for all customers, and we set $M = L$. However, if $\sum_{i=1}^{L} \lambda_i < \Lambda$, then the arrival rate of $\lambda_{L+1} := (\Lambda - \sum_{i=1}^{L} \lambda_i)$ essentially abandons the system because there is no capacity applied to processing it. In this case, we set $M = L + 1$ and consider these customers as class $(L + 1)$ customers, and this class is not allocated any capacity; hence, the corresponding offered wait is ∞. Thus, for any policy $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$, we can construct another class-based policy that we denote by $(\lambda(\kappa, v), w(\kappa, v))$ with $\lambda_i(\kappa, v) := \lambda_i$, where λ_i satisfies (3) for $i = 1, \ldots, M$, $w_i(\kappa, v) := v_i$ for $i = 1, \ldots, L$, and if $M = L + 1$, we have $w_{L+1}(\kappa, v) = \infty$.

Before formally proving that this class-based policy is equivalent to the original policy, we illustrate this equivalence in the previously discussed example of a policy that applies capacities κ_1 and κ_2 at times v_1 and v_2, respectively. This policy has the workload profile given by Figure 2(a). In the class-based view, we divide the arriving work into three classes, with arrival rates λ_1, λ_2, and λ_3. The classes 1 and 2 are allocated capacities κ_1 and κ_2 resulting in waits $w_1 = v_1$ and $w_2 = v_2$. The class 3 customers simply abandon the system, so that $w_3 = \infty$. This is depicted in Figure 2(b). Note that if we combine the workload profiles for the three classes (by simply adding them up), then we obtain the workload profile for the overall system. This equivalence is indeed true in general and is stated formally in the following result.

Lemma 1. *The workload profile for any feasible policy $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$ is identical to that for the corresponding class-based policy $(\lambda(\kappa, v), w(\kappa, v))$.*

We will soon see that this class-based approach is extremely convenient to analyze. In §5.1.2, we show how the optimal policy can be implemented by prioritizing customers on the basis of their time in the system without creating any classes.

3.2. Characterizing System Cost

We next formalize the costs incurred by the system. These costs depend on the time customers spend waiting in the system and on whether they ultimately were able to obtain service. In particular, these costs depend on the customer's patience time and the delay after which the customer is expected to be processed (the offered wait). To compute this system cost, we first define a customer cost function $\xi\colon \mathbb{R}_+^2 \to \mathbb{R}$ that is a function of the customer's patience time and the offered wait. Specifically, $\xi(y, w)$ is the cost incurred by the system due to a unit mass of customer whose patience time is y and whose offered wait is w.

Using this definition, for an FCFS policy that has an offered wait of $\bar{w}$, a unit of customer mass that arrives with a patience time $y \in [0, \infty)$ has a customer cost of $\xi(y, \bar{w})$, and thus the total system cost under FCFS equals

$$\Lambda \int_0^\infty \int_0^\infty \xi(y, \bar{w}) f(x, y)\, dy\, dx.$$

For non-FCFS policies, unlike FCFS policies, the offered wait is not the same for all customers. So, to compute the total system cost, we need to aggregate over all the potential offered waits possible. For each possible offered wait w, we define

$$c(w) = \int_0^\infty \int_0^\infty \xi(y, w) f(x, y)\, dy\, dx, \tag{4}$$

which should be interpreted as the steady-state cost per customer when the offered wait is w. We assume that $c'(w) = (d/dw)c(w) \geq 0$, for all $w \geq 0$. Thus, using the class-based approach, we can characterize the steady-state cost associated with a policy (κ, v) by

computing the cost under FCFS for each class and then summing it up; that is, we have

$$C(\kappa, v) = \sum_{i=1}^{|\lambda(\kappa, v)|} \lambda_i(\kappa, v) c(w_i(\kappa, v)), \tag{5}$$

where $\lambda_i(\kappa, v)$ and $w_i(\kappa, v)$ are the arrival rate and offered wait for class i in the class-based representation for $i = 1, 2, \ldots, M$ (as discussed in §3.1.2).

Remark 1. Notice that because our focus is on the fluid steady state, the above cost characterization works for any policy that leads to the fluid steady-state workload $\eta(t)$. There may be policies for which the workload profile does not correspond to a $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$, but instead requires infinite classes. For such a case, we can use the underlying continuity to approximate the cost function to any desired accuracy using $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$. In this sense, our focus on the set of policies $(\mathcal{K}, \mathcal{V})$ is not that restrictive.

3.2.1. Three Performance Metrics. We next use different functional forms for ξ to obtain three key performance metrics: the long-run average rate of abandonment, the expected steady-state queue length, and the offered wait. For each case, we characterize the steady-state cost per customer function c. The total cost for any policy can then be obtained using (5).

Abandonment Metric. If the offered wait is w, then any arriving customer with patience time $y < w$ would abandon the system before obtaining service, whereas those with patience times $y \geq w$ would eventually get served. So, defining $\xi(y, w) = \mathbb{I}\{y < w\}$, we obtain the probability that a customer abandons as

$$c(w) = \int_0^\infty \int_0^\infty \mathbb{I}\{y < w\} f(x, y)\, dy\, dx = F_A(w), \tag{6}$$

where $F_A(w)$ represents the marginal cumulative distribution of patience times. Thus, using $c(w)$ in (5) we can calculate the long-run average rate of customer abandonment for any policy.

Queue Length Metric. We compute the expected steady-state waiting time in queue incurred by each customer in the system and then apply Little's law to obtain the queue length metric. Noting that if the offered wait is w, then any arriving customer with patience time y would wait $\xi(y, w) = \min\{y, w\}$ time units in the system, we compute the expected steady-state waiting time in the queue for an arriving customer as

$$\begin{aligned} c(w) &= \int_0^\infty \int_0^\infty \min\{y, w\} f(x, y)\, dy\, dx \\ &= \int_0^w y f_A(y)\, dy + w\bar{F}_A(w) \\ &= \int_0^w \bar{F}_A(y)\, dy. \end{aligned} \tag{7}$$

Thus, using $c(w)$ in (5), we obtain the overall expected steady-state queue length.

Offered Wait Metric. The offered wait for a customer is trivially obtained by setting $\xi(y, w) = w$ so that $c(w) = w$. Notice that using this in (5) for any policy yields an aggregated version of the offered wait across all customers. If we divide this aggregated cost by the overall arrival rate Λ, then we obtain the expected steady-state offered wait. Since, this expectation is of more interest, we can directly obtain it by setting $c(w) = w/\Lambda$.

4. Optimizing Fluid Performance

In this section, we proceed with our analysis by optimizing the system cost in the fluid model. We derive the optimal fluid solution in §4.1. Then, in §4.2, we focus on the case in which service and patience times are independent to provide more insight into the optimal policies.

4.1. Characterizing Optimal Policies

Our objective is to minimize the total expected steady-state system cost by selecting the policy (κ, v). In particular, we wish to solve

$$\inf_{(\kappa, v) \in (\mathcal{K}, \mathcal{V})} C(\kappa, v). \tag{8}$$

Using the equivalent view explained in the previous section, we would like to use the class-based approach for the optimization problem. In particular, we would like to state this optimization problem as one of finding the optimal number of classes to split the customers into, along with the arrival rate and offered wait for each class. Formally, we define the set of feasible policies in the space of customer classes as $(\mathcal{L}, \mathcal{W})$ so that $(\lambda, w) \in (\mathcal{L}, \mathcal{W})$ if $(\lambda, w) \in \mathbb{R}_+^M \times \bar{\mathbb{R}}_+^M$ for some $M \in \mathbb{N}_+$, where $\bar{\mathbb{R}}_+ = \mathbb{R} \cup \{\infty\}$, such that $w_i < w_j$ for $i < j$ (w_M may equal ∞), $\sum_{i=1}^M m\lambda_i \phi(w_i) \leq n$, and $\sum_{i=1}^M \lambda_i = \Lambda$. Thus, a class-based approach entails solving the following optimization problem:

$$\inf_{(\lambda, w) \in (\mathcal{L}, \mathcal{W})} \sum_{i=1}^{|\lambda|} \lambda_i c(w_i). \tag{9}$$

We next prove that the optimization problems (8) and (9) are equivalent. For this, it suffices to prove that for any feasible class-based policy $(\lambda, w) \in (\mathcal{L}, \mathcal{W})$, we can construct a feasible policy $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$ that has identical cost. The following result formally proves this.

Lemma 2. *For any feasible class-based policy $(\lambda, w) \in (\mathcal{L}, \mathcal{W})$, there exists a feasible policy $(\kappa, v) \in (\mathcal{K}, \mathcal{V})$ with an identical workload profile and, further, identical cost.*

Thus, Lemmas 1 and 2 imply that the optimization problem (9) is equivalent to (8) in the sense they have identical optimal objective function values and a policy that solves (9) can be mapped to a policy

that solves (8). Thus, henceforth, we focus on the optimization problem (9), and we begin our analysis by noting that this optimization problem can be written out as follows:

$$\begin{aligned} \inf_{M\in\mathbb{N};\,(\lambda,\,w)\in\mathbb{R}_+^M\times\bar{\mathbb{R}}_+^M} \quad & \sum_{i=1}^{M}\lambda_i c(w_i) \\ \text{s.t.} \quad & \sum_{i=1}^{M}\lambda_i\phi(w_i)\le\frac{n}{m}, \\ & \sum_{i=1}^{M}\lambda_i=\Lambda, \\ & 0\le w_1<w_2<\cdots<w_M\le\infty. \end{aligned} \tag{10}$$

The following proposition proves a nice property of this optimization problem that will simplify our analysis considerably.

Proposition 1. *There exists an optimal solution to* (10) *that creates at most two classes; that is, there exists an optimizer of* (10), λ^*, *with* $|\lambda^*|\le 2$.

Proposition 1 proves that the optimal solution to(10) is given by the solution to the following:

$$\begin{aligned} \inf_{0\le\lambda_\ell,\,\lambda_h;\,0\le w_\ell<w_h\le\infty} \quad & \lambda_\ell c(w_\ell)+\lambda_h c(w_h) \\ \text{s.t.} \quad & \lambda_\ell\phi(w_\ell)+\lambda_h\phi(w_h)\le\frac{n}{m}, \\ & \lambda_\ell+\lambda_h=\Lambda. \end{aligned} \tag{11}$$

Because $\Lambda > n/m$, it suffices to consider solutions that utilize all the capacity.[3] Writing $\lambda_h=\Lambda-\lambda_\ell$, then we can simplify (11) to

$$\inf_{0\le\lambda_\ell\le\Lambda;\,0\le w_\ell<w_h\le\infty} \quad \lambda_\ell c(w_\ell)+(\Lambda-\lambda_\ell)c(w_h) \tag{12}$$

$$\text{s.t.} \quad \lambda_\ell\phi(w_\ell)+(\Lambda-\lambda_\ell)\phi(w_h)=\frac{n}{m}. \tag{13}$$

We next use $\bar{w}$, the offered wait for the FCFS policy, to further simplify this optimization problem. Because $\Lambda\phi(\bar{w})=n/m$, for (13) to hold, we cannot have $w_\ell, w_h<\bar{w}$ or $w_\ell, w_h>\bar{w}$. It follows that we can restrict attention to (w_ℓ, w_h) that satisfy $0\le w_\ell\le\bar{w}<w_h\le\infty$. For such a pair of waiting times, we can compute λ_ℓ explicitly as a function of (w_ℓ, w_h) as follows:

$$\lambda_\ell(w_\ell, w_h)=\frac{n-m\Lambda\phi(w_h)}{m(\phi(w_\ell)-\phi(w_h))}. \tag{14}$$

[3] In case a solution exists that does not utilize the entire capacity and so the capacity constraint holds with a strict inequality, then because ϕ and $c(w)$ are both strictly decreasing continuous functions, we can obtain a lower cost solution by decreasing either w_ℓ or w_h by a small amount. Thus, we obtain a contradiction to the initial assumption that the optimizer does not utilize the entire capacity.

Furthermore, because $w_\ell\le\bar{w}<w_h$, we have $\Lambda\ge\lambda_\ell(w_\ell, w_h)\ge 0$. This implies (12)–(13) can be simplified further as follows:

$$\inf_{0\le w_\ell\le\bar{w}<w_h\le\infty}\lambda_\ell(w_\ell, w_h)c(w_\ell)+(\Lambda-\lambda_\ell(w_\ell, w_h))c(w_h). \tag{15}$$

Thus, the optimal solution to the scheduling problem is fully characterized by a pair of offered waiting times (w_ℓ, w_h) that solves (15).

To solve (15), it is useful to first characterize the different types of solutions possible. The first candidate is the FCFS policy, which corresponds to creating a single class of customers with an offered wait of $\bar{w}$. This policy can be represented is multiple ways: either as $(\bar{w}, w_h)$ for any $w_h>\bar{w}$, because then $\lambda_\ell(\bar{w}, w_h)=\Lambda$ and $\lambda_h(\bar{w}, w_h)=0$, or as $(w_\ell, \bar{w})$ for any $0\le w_\ell<\bar{w}$, because then $\lambda_\ell(w_\ell, \bar{w})=0$ and $\lambda_h(w_\ell, \bar{w})=\Lambda$.

The second candidate for optimality is one with offered waiting times of $(0,\infty)$, that is, one class of customers is served immediately with an offered waiting time of zero, whereas the other class is not allocated any capacity, and all these customers eventually abandon. In the overloaded regime, this second candidate is equivalent to operating the original (fluid) system in a single class under an LCFS policy. This is so because under LCFS, the system processes work at rate n from the incoming work at every time instant, and hence this work receives an offered wait of zero, whereas the remaining work $m\Lambda-n$ is never processed, and hence receives an offered wait of infinity.

In addition to the above described candidate policies, we have three additional types of candidate policies possible:

1. positive and finite offered waits for both customer classes, that is, policies of the form (w_ℓ, w_h) where $0<w_\ell<\bar{w}<w_h<\infty$;

2. zero offered wait for one class and a finite offered wait for the other class, that is, policies of the form $(0, w_h)$, where $\bar{w}<w_h<\infty$; and

3. positive and finite wait for one class and an infinite offered wait for the other class (this class is not allocated any capacity), that is, policies of the form (w_ℓ, ∞), where $0<\bar{w}_\ell<\bar{w}$.

The above candidates cover the entire set of feasible policies $\{(w_\ell, w_h)\colon 0\le w_\ell\le\bar{w}<w_h\le\infty\}$.

Next, we refine the set of candidates of the form (w_ℓ, w_h), $(0, w_h)$, and (w_ℓ, ∞) using the first-order necessary conditions for optimality as follows:

Proposition 2. 1. *An optimal solution to* (15) *of the form* (w_ℓ, w_h) *with* $0<w_\ell<\bar{w}<w_h<\infty$ *must satisfy*

$$\frac{\phi'(w_\ell)}{c'(w_\ell)}=\frac{\phi(w_h)-\phi(w_\ell)}{c(w_h)-c(w_\ell)}=\frac{\phi'(w_h)}{c'(w_h)}. \tag{16}$$

2. *An optimal solution to* (15) *of the form* $(0, w_h)$ *with* $\bar{w} < w_h < \infty$ *must satisfy*

$$\frac{\phi(w_h) - 1}{c(w_h) - c(0)} = \frac{\phi'(w_h)}{c'(w_h)}. \tag{17}$$

3. *An optimal solution to* (15) *of the form* (w_ℓ, ∞) *with* $0 < w_\ell < \bar{w}$ *must satisfy*

$$\frac{\phi'(w_\ell)}{c'(w_\ell)} = \frac{-\phi(w_\ell)}{c(\infty) - c(w_\ell)}. \tag{18}$$

Policies characterized in Proposition 2 along with FCFS and LCFS are *all* the possible candidates for optimality. We next present some special cases under which we can explicitly identify the optimal policy. To do so, we observe from Proposition 2 that the first-order conditions for optimality depend on the following term:

$$r(w) := -\frac{\phi'(w)}{c'(w)} \quad \text{for } w \geq 0.$$

It will be useful to relate this function to the optimization problem (11). The function $-\phi'(w)$ measures the rate at which additional capacity becomes available as we make the customers wait more, and in this sense relates to the capacity constraint. Similarly, $c'(w)$ measures the rate at which the cost or the objective function increases as we make the customers wait more. Given that we have a fixed capacity constraint, it follows that the ratio $r(w) = -\phi'(w)/c'(w)$ captures the "bang for the buck" in this setting.

We next prove that if r is monotone, then none of the first-order conditions listed in Proposition 2 can hold, which implies that the optimal policy for this case must be either FCFS or LCFS. The following proposition provides the formal result.

Proposition 3. 1. *If* r *is decreasing, then the optimal policy is FCFS.*

2. *If* r *is increasing, then* $(0, \infty)$ *is the optimal solution to* (15)*, i.e., the optimal policy is LCFS.*

When r is nonmonotone, other policies listed in Proposition 2 may be optimal. We will investigate these in detail in the next section for the case in which service times and patience times are independent, where we will also observe that the function r will take more familiar forms, such as the hazard rate or the density of the patience time distribution. We discuss the case of dependent service and patience times further in Appendix B.

4.2. Independent Service and Patience Times

In this section, we focus on the case in which the service requirement of the customers is independent of their patience level; that is,

$$f(x, y) = f_S(x) f_A(y),$$

where f_S and f_A correspond to the marginal distributions for service and patience times, respectively. In this case, the workload depletion function ϕ is given by

$$\phi(w) = \bar{F}_A(w) = \int_w^\infty f_A(w)\, dw. \tag{19}$$

This relation can also be understood as follows: for every unit of arriving customer mass, the fraction of customers that remain in the system after w time units is given by $\bar{F}_A(w)$. Because the customers' work requirements are independent of their patience, it follows that the fraction of remaining work is also equal to $\bar{F}_A(w)$.

We next consider our three performance metrics and characterize the optimal policy for each metric.

4.2.1. Total Abandonment Rate Metric. We can compute the cost based on an offered wait of w as in (6), i.e., we have $c(w) = F_A(w)$. So, the optimization problem (11) becomes

$$\inf_{0 \leq \lambda_\ell, \lambda_h;\, 0 \leq w_\ell < w_h \leq \infty} \quad \lambda_\ell F_A(w_\ell) + \lambda_h F_A(w_h) \tag{20}$$

$$\text{s.t.} \quad \lambda_\ell \bar{F}_A(w_\ell) + \lambda_h \bar{F}_A(w_h) = \frac{n}{m}, \tag{21}$$

$$\lambda_\ell + \lambda_h = \Lambda. \tag{22}$$

Noting that $F_A = 1 - \bar{F}_A$, it follows that the objective value of this optimization problem is a constant and equals $(\Lambda - n/m)$ over the entire feasible set; that is, all nonidling policies (that utilize all the available capacity) have the same cost. The intuition behind this result is as follows: any nonidling policy can only process customers at rate n/m because the service time is independent of the patience time. Thus, since the arrival rate is Λ, the total abandonment rate is $\Lambda - n/m$.

Proposition 4. *If service times and patience times are independent, then the total abandonment rate for any nonidling policy, i.e., that satisfies the constraints* (21) *and* (22)*, equals* $(\Lambda - n/m)$.

We would like to point out that if the service times and patience times were dependent, then the above result would not hold. This is so because due to the dependence between the service and patience times, these policies can differentiate between customers with different service times and hence can differ in the number of customers processed per unit time. See §5.2.2 for a numerical study that illustrates that for the case of dependent service and patience times, the optimal policy can outperform FCFS significantly on the abandonment rate metric.

4.2.2. Queue Length Metric. In this case, using (7), we have $c(w) = \int_0^w \bar{F}_A(y)\,dy$. Thus, the optimization problem (11) becomes

$$\begin{aligned} \inf_{0\le\lambda_\ell,\,\lambda_h;\,0\le w_\ell<w_h\le\infty} \quad & \lambda_\ell \int_0^{w_\ell} \bar{F}_A(y)\,dy + \lambda_h \int_0^{w_h} \bar{F}_A(y)\,dy \\ \text{s.t.} \quad & \lambda_\ell \bar{F}_A(w_\ell) + \lambda_h \bar{F}_A(w_h) = \frac{n}{m}, \\ & \lambda_\ell + \lambda_h = \Lambda. \end{aligned} \tag{23}$$

We also have

$$r(w) = -\frac{\phi'(w)}{c'(w)} = \frac{f_A(w)}{\bar{F}_A(w)};$$

that is, $r(w) = h_A(w)$, which is the hazard rate of the patience time distribution. Using Proposition 3, we obtain the following result for the setting where the patience time distribution has a monotone hazard rate:

Corollary 1 (Monotone Hazard Rates). 1. *If the hazard rate of the patience time distribution is decreasing, then FCFS minimizes the queue length.*

2. *If the hazard rate of the patience time distribution is increasing, then LCFS minimizes the queue length.*

The intuition behind this result is as follows. Consider the case in which the hazard rate is decreasing. Then, from the system's perspective, as customers wait in the queue, their hazard rate decreases, which implies that they are more willing to wait in the queue. It follows that to minimize the queue length we should focus on processing customers who would otherwise continue waiting in the queue (and would not abandon). Thus, we obtain that FCFS is optimal. Notice that because of Proposition 4, by serving customers in an FCFS manner, we have the lowest queue length while keeping the total abandonment rate constant. Analogously, if the hazard rate is increasing, then the LCFS policy minimizes the queue length while keeping the total abandonment rate constant. Note that if the hazard rate is a constant, i.e., patience times are exponentially distributed, then it is easy to verify that any nonidling policy would have the same queue length, and thus, in this case, both FCFS and LCFS are trivially optimal.

The next result characterizes the optimal policy for the case of hazard rate functions that are quasi-concave and nonmonotone. In this case, (17) is the only first-order condition of those listed in Proposition 2 that may hold. Defining $\gamma = 1/\int_0^\infty y f_A(y)\,dy$ so that $1/\gamma$ is the mean patience time, we have the following result:

Proposition 5 (Quasi-Concave Hazard Rates). *If the hazard rate of the patience time distribution is quasi-concave and nonmonotone, then we have the following*:

1. *If $h_A(\infty) \ge \gamma$, then LCFS is optimal.*

2. *If $h_A(\infty) < \gamma$, then there is a unique solution $w = w_h \in (0, \infty)$ to*

$$h_A(w) = \frac{F_A(w)}{\int_0^w \bar{F}_A(y)\,dy}, \tag{24}$$

and if this solution $w_h > \bar{w}$, then $(0, w_h)$ is the optimal solution to (23); otherwise, FCFS is the optimal solution.

Corollary 1 and Proposition 5 provide a simple characterization of the optimal policy that minimizes the expected queue length for patience distributions with hazard rate functions that are either monotone or quasi-concave. In general, when hazard rate functions need not have such properties, any of the policies described in §4.1 could be optimal. If the hazard rate function is quasi-convex and nonmonotone, then an analogous result to Proposition 5 is obtained with an optimal policy being of the form (w_ℓ, ∞), where w_ℓ is the unique solution to (18), FCFS, or LCFS.

4.2.3. Offered Wait Metric. We next consider the objective of minimizing the expected steady-state offered wait. In this case, we have $c(w) = w/\Lambda$, and so the optimization problem (11) becomes

$$\begin{aligned} \inf_{0\le\lambda_\ell,\,\lambda_h;\,0\le w_\ell<w_h\le\infty} \quad & \frac{\lambda_\ell}{\Lambda} w_\ell + \frac{\lambda_h}{\Lambda} w_h \\ \text{s.t.} \quad & \lambda_\ell \bar{F}_A(w_\ell) + \lambda_h \bar{F}_A(w_h) = \frac{n}{m}, \\ & \lambda_\ell + \lambda_h = \Lambda. \end{aligned} \tag{25}$$

Note that when minimizing offered wait, LCFS can never be optimal. This is because under LCFS, a portion of arriving customers obtain an offered wait of ∞.

To characterize the optimal policy, we begin by computing r for this setting:

$$r(w) = -\frac{\phi'(w)}{c'(w)} = f_A(w),$$

the density of the patience time distribution. As in the previous section, we will consider the cases in which r is monotone or unimodal. Using Proposition 3, we obtain the following result for the case in which the patience time distribution has monotone decreasing density. (Trivially, we cannot have monotone increasing density.)

Corollary 2 (Monotone Density). *If the density of the patience time distribution is decreasing, then FCFS minimizes the offered wait.*

It is interesting to consider the case of exponentially distributed patience times. For this distribution, we have seen that all nonidling policies have identical performance for the queue length metric, but when dealing with the offered wait metric, it turns out that different policies may perform differently,

and the above result implies that FCFS in fact minimizes the offered wait. This difference stems from the fact that because of customer abandonments, the relation between the expected offered wait and queue length is intricate (unlike the case without abandonments where Little's law relates these quantities); for instance, although the LCFS policy and FCFS have the same expected steady-state queue length, under LCFS the expected offered wait will be infinite.

The next result characterizes the optimal policy for the case in which the density is quasi-concave and nonmonotone.

Proposition 6 (Quasi-Concave Density). *If the density of the patience time distribution is quasi-concave and nonmonotone, then there is a unique solution* $w = w_h \in (0, \infty)$ *to*

$$f_A(w) = \frac{F_A(w)}{w}, \tag{26}$$

and if this solution $w_h > \bar{w}$, *then* $(0, w_h)$ *is the optimal solution to* (25); *otherwise FCFS is the optimal solution.*

This result is analogous to Proposition 5 with the exception that LCFS cannot be optimal in this case. Similar to the case of the queue length metric, it is not possible to characterize the optimal policy for general density functions. However, for the offered wait metric, some policies cannot be optimal; in particular, policies of the form (w_ℓ, ∞) cannot be optimal because these lead to infinite offered waits. This implies that the candidate policies to consider are FCFS, (w_ℓ, w_h), and $(0, w_h)$. Noting that the latter two types of policies require w_h to exceed the FCFS offered wait $\bar{w}$, it follows that for high arrival rates, these policies would be infeasible, and the only policy that can be optimal is FCFS. This is formally stated in the following result.

Corollary 3. *For the offered wait metric, if all solutions to* (16) *and* (17) *are bounded, then FCFS is optimal for all arrival rates* $\Lambda \geq \bar{\Lambda}$ *for some finite threshold* $\bar{\Lambda}$.

4.2.4. Fluid Optimal Policies for Different Patience Distributions. We next discuss how our findings apply to three classes of distributions that have been used in the literature to model customer abandonments, namely, lognormal, Erlang-3, and hyperexponential distributions. (The Erlang and hyperexponential distributions were used in Jouini et al. 2013, and Mandelbaum and Zeltyn 2013 used mixtures of lognormal distributions to fit the abandonment distribution.) Table 1 states the optimal control policy for each class of distributions for the queue length and offered wait metrics. Notice that for hyperexponential distributions, the hazard rate and density are both decreasing, and hence FCFS is always the optimal policy. For Erlang-3 distribution, the hazard rate is monotone increasing, and thus LCFS minimizes the queue length, and the density is quasi-concave and nonmonotone; hence, the optimal policy is either FCFS or of the form $(0, w_h)$. Lognormal distributions have quasi-concave and nonmonotone hazard rate and density functions, and $\lim_{w\to\infty} h_A(w) = 0$. Hence, for the lognormal distribution, the optimal policies to minimize the queue length and offered wait metrics are either FCFS or of the form $(0, w_h)$. Finally, the table also lists the optimal policy for exponential patience times; in this case, as explained previously, the expected queue length is identical for all nonidling policies, and FCFS minimizes the offered wait.

Table 1 Optimal Policy for Different Classes of Abandonment Distributions for Minimizing Queue Length and Offered Wait Performance Metrics

	Optimal policy	
Distribution	Queue length	Offered wait
Lognormal	$(0, w_h)$ or FCFS	$(0, w_h)$ or FCFS
Erlang-3	LCFS	$(0, w_h)$ or FCFS
Hyperexponential	FCFS	FCFS
Exponential	Any nonidling policy	FCFS

5. Applying Fluid-Based Insights to Queueing Systems

Our approach thus far has focused on the fluid model. In this section, we apply the insights we have obtained to actual queueing systems. In particular, in §5.1, we propose two policies that implement the fluid prescription. In §5.1.1, we describe a naive implementation of the fluid solution that partitions the capacity. Given that such a partitioning is inefficient, in §5.1.2, we propose a time-in-queue policy that dynamically prioritizes customers without partitioning capacity. In §5.2, we show that the proposed TIQ policy dominates FCFS, and finally, in §5.3, we discuss the accuracy of the fluid model in approximating queueing systems.

5.1. Implementing the Fluid Prescription

5.1.1. A Naive Prescription: Creating Two Classes. The naive prescription implements any fluid solution (w_ℓ, w_h) by splitting the customers into two separate classes and by partitioning the servers into two different pools so that each server pool processes one distinct customer class. This policy splits the arrival rate into two classes with arrival rates (λ_1, λ_2), where $\lambda_1 := \lambda_\ell(w_\ell, w_h)$, as defined in (14), and $\lambda_2 := \Lambda - \lambda_1$; that is, an arriving customer is allocated to class 1 with probability λ_1/Λ, and to class 2 otherwise. The server pool is also split for the two classes; in particular, class 1 is assigned $n_1 := \lambda_1 \bar{F}(w_\ell)$ servers, and class 2 is assigned $n_2 := n - n_1$ servers.

5.1.2. A TIQ Policy: Dynamically Prioritizing Customers. We next propose a policy that implements the fluid solution (w_ℓ, w_h) without splitting the customers, but by dynamically prioritizing them based on their time spent in the queue. In this sense, we refer to this policy as the time-in-queue policy. The key idea behind the policy is to ensure that in the fluid model's steady state, the customers who get served in the system have a waiting time of either w_ℓ or w_h time units. This implies that the capacity is applied at the offered waits of w_ℓ and w_h, and thus, based on the analysis in §3.1.2, it follows that this policy would lead to the same steady-state workload profile as in the case in which the customers are divided into two classes with offered waits of w_ℓ and w_h, respectively. Thus, the performance metrics would also be the same.

This alternative policy, denoted by TIQ(w_ℓ, w_h), is defined via the following priority rule:

(a) First process any customer who has waited more than or equal to w_h time units, giving priority to the customer who has waited the longest among them (we apply FCFS among these customers).

(b) If there are no customers who have waited more than w_h time units, then process customers who have waited less than w_ℓ, giving priority to the customer who has waited longest among them (we apply FCFS among these customers).

(c) If there are no customers who have waited less than w_ℓ or more than w_h, then process the remaining customers, giving priority to the one who has waited least (we apply LCFS among these customers).

Note that because $w_h > \bar{w}$, we have $m\Lambda\bar{F}(w_h) < n$, and thus step (a) of the policy ensures that in steady state there will not be any fluid mass in the system waiting for longer than w_h time units. Similarly, because $w_\ell < \bar{w}$, we have $m\Lambda\bar{F}(w_\ell) > n$, and step (b) ensures that in steady state there will not be any fluid mass in the system waiting for less than w_ℓ time units. In fact, these steps ensure that the capacity is applied at the offered waits of w_ℓ and w_h so that the customers in the system wait for either w_ℓ or w_h time units before being served.

Step (c) plays a subtle role in this policy. Notice that step (c) plays a role only if $w_\ell = 0$ because in this case step (b) will never be applied in the actual queueing system. If $w_\ell = 0$, then we need to ensure that the policy in step (c) leads us to the desired fluid steady state of $(0, w_h)$. We know from the previous section that an overall LCFS policy leads to the steady state of $(0, \infty)$ in the overloaded fluid system. So, when we apply LCFS in step (c) to customers whose time-in-queue is between 0 and w_h, because the system is overloaded, in steady state, all customers that are actually processed will be processed at a time-in-queue of 0, and others will simply wait until their time-in-queue exceeds w_h, at which point these customers will be served when step (a) is applied. Hence, we achieve the $(0, w_h)$ fluid solution.

Suppose, instead of applying LCFS, we implemented an FCFS policy in step (c) for the case $(0, w_h)$. Then, if we consider the corresponding fluid dynamics starting from an empty system, the resulting steady state would be as if *all* customers were served in an FCFS manner, because asymptotically customers would have FCFS wait times of $\bar{w}$, which is less than w_h. Thus, for the case of $w_\ell = 0$, the choice of LCFS in step (c) is important. However, if $w_\ell > 0$, then step (c) is not relevant, and one could even idle any available servers remaining after applying steps (a) and (b) in the queueing system. However, given that nonidling policies are expected to perform better and the fact that the use of LCFS in step (c) is beneficial for the case $w_\ell = 0$, we propose using it in general.

Relation Between FCFS and TIQ Policies. Notice that if $w_h < \bar{w}$, the TIQ policy essentially operates in an FCFS manner. This is so because, in this overload system, we have $m\Lambda\bar{F}(w_h) > m\Lambda\bar{F}(\bar{w}) = n$, and hence, in steady state the TIQ policy will reduce to only implementing step (a), which processes customers in an FCFS manner.

Robustness of TIQ Policy to Changes in Arrival Rate. To implement the TIQ policy, we only need to know the fluid optimal solution (w_ℓ, w_h). This is in contrast to the naive policy, in which the splitting of customers requires the additional knowledge of the arrival rate. Thus, noting that the fluid solution is either FCFS, LCFS, or satisfies the arrival-rate-independent conditions in Proposition 2, we expect the TIQ policy to be optimal even for small changes in the arrival rate. In fact, it is possible that for some cases, the proposed TIQ policy may even be optimal for *all* arrival rates. This is so for the case of minimizing the queue length for quasi-concave and nonmonotone hazard rates. We will show this by applying Proposition 5. First, consider the case that $h_A(\infty) < \gamma$. Then, the optimal policy is either $(0, w_h)$, where w_h solves (24), or FCFS, and further, the optimal policy is $(0, w_h)$ if and only if $w_h > \bar{w}$. Noting that if $w_h < \bar{w}$, then the policy $TIQ(0, w_h)$ becomes equivalent to FCFS, it follows that the optimal policy is $TIQ(0, w_h)$ for all arrival rates. Next consider the case $h_A(\infty) \geq \gamma$; then the optimal policy is LCFS or $TIQ(0, \infty)$ for all arrival rates. So, for patience distributions with quasi-concave hazard rate functions, the policy that minimizes the queue length metric is indeed arrival rate independent.

5.2. Performance of Proposed Policies

We next numerically study the performance of our proposed naive and TIQ policies compared with that of the FCFS policy in queueing systems.

Table 2 A Comparison of the Expected Steady-State Queue Lengths Under the FCFS, Naive, and TIQ Policies

	$\rho = 1.05$			$\rho = 1.1$			$\rho = 1.5$		
		Prescription			Prescription			Prescription	
Arrival rate Λ	FCFS	Naive	TIQ	FCFS	Naive	TIQ	FCFS	Naive	TIQ
Lognormal									
25	15.4	+4%	−26%	19.3	+2%	−25%	40.2	+3%	−8%
50	26.3	−4%	−37%	35.0	−7%	−34%	77.1	−0%	−10%
100	48.2	−14%	−46%	71.5	−15%	−39%	154.2	−3%	−11%
500	249.5	−33%	−58%	347.7	−28%	−45%	761.0	−6%	−12%
Erlang-3									
25	21.9	−22%	−53%	26.8	−27%	−53%	46.3	−24%	−38%
50	39.3	−29%	−62%	50.8	−35%	−61%	90.7	−28%	−42%
100	74.9	−38%	−69%	104.4	−42%	−65%	181.8	−31%	−43%
500	390.9	−54%	−78%	513.9	−53%	−71%	903.8	−36%	−44%

Note. The TIQ policy has the best performance.

5.2.1. Independent Service and Patience Times. We consider two patience distributions: lognormal distribution and Erlang-3. The lognormal distribution we use is such that the natural logarithm of this random variable is normally distributed with unit mean and unit variance (this is obtained by setting the mean and variance of the lognormal distribution as $e^{3/2}$ and $(e^4 - e^3)$, respectively). The Erlang-3 distribution we use has mean equal to 3, i.e., each phase has unit mean. We study the performance metrics of expected steady-state queue length and offered wait in $M/M/n + GI$ systems with arrival rates of $\Lambda = 25, 50, 100$, and 500. For each arrival rate, we consider three system load values $\rho = 1.05$, 1.1, and 1.5 and set the number of servers $n = \lfloor m\Lambda/\rho \rfloor$; we also set $m = 1$. We computed the FCFS and naive policy performance metrics using the exact formulae in Zeltyn and Mandelbaum (2005). For the TIQ policy, we computed the performance metrics by simulating the queueing system for 10,000 time units and computing an average over 20 such simulations; the confidence interval half-width was less than 2.5% of the average reported performance metric in all cases. The results are displayed in Tables 2 and 3. In each table, we report the performance of FCFS in absolute terms and the performance of the naive and TIQ policies relative to the FCFS performance (improvements are indicated by negative values).

Consider Table 2, which compares the performance of the expected steady-state queue lengths. We observe that the TIQ policy performs better than the FCFS policy for all parameters. Furthermore, the TIQ policy dominates the naive policy as expected, because it does not idle the servers unnecessarily. In fact, we observe that in one case, the naive policy performs worse than the FCFS policy. This occurs for systems in which customer patience is lognormally distributed and the system volume is the smallest among those that we have considered, i.e., $\Lambda = 25$. An explanation for this is that the split of capacity that occurs in the naive policy leads to an inefficiency that we expect to be manifested on the order of

Table 3 A Comparison of the Expected Steady-State Offered Waits Under the FCFS, Naive, and TIQ Policies

	$\rho = 1.05$			$\rho = 1.1$			$\rho = 1.5$		
		Prescription			Prescription			Prescription	
Arrival rate Λ	FCFS	Naive	TIQ	FCFS	Naive	TIQ	FCFS	Naive	TIQ
Lognormal									
25	0.65	+17%	−15%	0.82	+14%	−13%	1.93	+4%	−0%
50	0.54	+7%	−25%	0.73	+5%	−22%	1.82	+3%	−1%
100	0.49	−3%	−35%	0.74	−3%	−25%	1.81	+2%	−1%
500	0.51	−23%	−47%	0.72	−15%	−31%	1.77	+1%	−1%
Erlang-3									
25	0.91	−2%	−34%	1.13	−5%	−34%	2.14	+1%	−12%
50	0.81	−13%	−43%	1.05	−16%	−43%	2.06	−3%	−14%
100	0.76	−24%	−48%	1.08	−24%	−48%	2.06	−5%	−15%
500	0.79	−42%	−54%	1.06	−36%	−54%	2.04	−9%	−16%

square root of the system volume, and for small systems this effect is comparable to the fluid-scale component. Thus, the gain from the fluid optimization is counterbalanced by the inefficiencies introduced for small system sizes. However, as the system volume increases, we notice that the performance of the naive policy improves and is close to the performance of the TIQ policy.

Similar observations can be made in Table 3, which compares the performance of the expected offered waits. An additional observation that can be made in this table is for the case $\rho = 1.5$ for the lognormal patience distribution. In this case, the fluid solution is very close to FCFS, which is reflected in the performance of the TIQ policy. In this case, the inefficiency of the naive policy results in a slightly poor performance for all system sizes, though the inefficiency diminishes as expected as the system size increases.

5.2.2. Correlated Service and Patience Times. We next study the performance of our proposed policies for the case of correlated service and patience times using a simulation study. We use the correlation structure discussed in Reich (2012) and Mandelbaum and Zeltyn (2013). In particular, we assume customer patience is exponentially distributed with mean 7.5 minutes. The service time is lognormally distributed in the following manner: for a customer with patience w, the expected service time is the natural logarithm of a normally distributed random variable with mean $\mu(w)$ and standard deviation $\sigma = \frac{1}{2}$, where

$$\mu(w) = \log\left(\frac{23}{6}\left(\frac{6}{5} - e^{-(7/20)w}\right)\right) - \frac{\sigma^2}{2}.$$

For this service time distribution, a customer with patience of w time units has an expected service time of

$$\mathbb{E}[\text{Service time} \mid \text{Patience} = w] = \frac{23}{6}\left(\frac{6}{5} - e^{-(7/20)w}\right). \quad (27)$$

Thus, in this model, customers who are more patient have a higher mean service time. In this case, the overall mean service time across all customers is $m = 1{,}541/435 \approx 3.54$ minutes.

For this system, we can compute the workload depletion function,

$$\phi(w) = \frac{1}{m}\int_{x=0}^{\infty}\int_{y=w}^{\infty} x f(x,y)\,dy\,dx = \frac{145}{134}\left(\frac{6}{5} - \frac{8}{29}e^{-(7/20)w}\right).$$

We will focus on the abandonment rate metric for this study. Using Proposition 2, we obtain that the fluid solution is $(0, \infty)$, which translates to a TIQ policy of LCFS. We next simulate this queueing system under FCFS, the naive policy, and the TIQ policy of LCFS for the arrival rates $\Lambda = 25, 50, 100$, and 500 (all rates are in customers per minute) with $\rho = 1.05$, 1.1, and 1.5. Table 4 displays the average fraction of abandonment under the FCFS policy for different arrival rates and system loads, along with those under the naive and TIQ policies that are stated relative to the FCFS performance. Each simulation was run for 30,000 time units and yielded a confidence interval half-width of less than 3.6% of the average reported performance metric in all cases. The results in the table show that in all cases, both naive and TIQ policies dominate FCFS by a significant amount.

5.3. Accuracy of Fluid Approximations

We next study the accuracy of the fluid model operating under naive and TIQ policies in approximating the actual queueing system. We first consider the case in which service and patience times are independent, and then move to the correlated case.

5.3.1. Independent Service and Patience Times. It is well known that for independent service and patience times, the fluid model provides a very good approximation for overloaded $M/M/n + GI$ systems operating under the FCFS policy (see Bassamboo and Randhawa 2010). We next study the asymptotic performance of applying the naive policy to an $M/M/n+GI$ system. To do so, we consider a high volume regime in which the arrival rate and the number of servers increase without bound while keeping all other system parameters constant. In this regime, we maintain a constant system load $\rho > 1$ so that for any arrival rate Λ, the number of servers equals $\lfloor m\Lambda/\rho \rfloor$. We denote the performance of the naive policy for

Table 4 A Comparison of the Average Fraction of Abandonment Under the FCFS, Naive, and TIQ Policies

	$\rho = 1.05$			$\rho = 1.1$			$\rho = 1.5$		
		Prescription			Prescription			Prescription	
Arrival rate Λ	FCFS	Naive	TIQ	FCFS	Naive	TIQ	FCFS	Naive	TIQ
25	0.128	−16%	−41%	0.197	−24%	−41%	0.462	−18%	−26%
50	0.127	−26%	−47%	0.193	−32%	−47%	0.462	−20%	−27%
100	0.123	−35%	−52%	0.194	−37%	−49%	0.462	−22%	−27%
500	0.125	−49%	−59%	0.195	−45%	−52%	0.463	−25%	−28%

Table 5 Illustration of Asymptotic Accuracy: Expected Queue Length Under the TIQ Policy Compared with the Corresponding Fluid Solution

	$\rho = 1.05$			$\rho = 1.1$			$\rho = 1.5$		
Arrival rate Λ	Actual	Fluid	Gap	Actual	Fluid	Gap	Actual	Fluid	Gap
Lognormal									
25	11.3	4.8	6.5	14.4	9.1	5.3	36.8	33.3	3.5
50	16.6	9.5	7.1	23.2	18.2	5.0	69.2	66.6	2.6
100	26.2	19.0	7.2	43.9	36.4	7.5	137.0	133.3	3.7
500	105.0	95.2	9.8	190.1	181.8	8.3	669.0	666.4	2.6
Erlang-3									
25	10.3	3.6	6.7	12.7	6.8	5.9	28.6	25.0	3.6
50	15.0	7.1	7.9	19.9	13.6	6.3	52.7	50.0	2.7
100	23.4	14.3	9.1	36.4	27.3	9.1	103.8	100.0	3.8
500	86.4	71.4	15.0	147.1	136.4	10.7	502.6	500.0	2.6

a system with arrival rate Λ in this sequence of $M/M/n+GI$ systems by $q^{\Lambda}_{N(w_\ell, w_h)}$ for the queue length metric and $w^{\Lambda}_{N(w_\ell, w_h)}$ for the offered wait metric, where the naive policy is denoted by $N(w_\ell, w_h)$, and we use the superscript Λ on the performance metrics to indicate the system volume.

Because the naive policy splits customers into two classes, each served in an FCFS manner, we can apply the results in Zeltyn and Mandelbaum (2005) and Bassamboo and Randhawa (2010) to characterize its asymptotic performance. In particular, those results directly yield that the performance of the naive policy is well approximated by the corresponding fluid value, which we denote by $q^{*,\Lambda}$ for the queue length metric and $w^{*,\Lambda}$ for the offered wait metric for a fluid system with arrival rate Λ; i.e., $q^{*,\Lambda}$ and $w^{*,\Lambda}$ are the optimal objective values of (23) and (25), respectively. We formally state this performance in the following result, where we use the standard order notation.[4]

Proposition 7. *For any $\rho > 1$, for an $M/M/n+GI$ queueing system with arrival rate Λ and number of servers $n = \lfloor m\Lambda/\rho \rfloor$, as Λ increases without bound, we have the following*:

1. *For the queue length metric, the naive policy $N(w_\ell, w_h)$, where (w_ℓ, w_h) solves (23), has the performance $q^{\Lambda}_{N(w_\ell, w_h)} = q^{*,\Lambda} + \mathcal{O}(\sqrt{\Lambda})$. Furthermore, if $w_\ell > 0$, we have $q^{\Lambda}_{N(w_\ell, w_h)} = q^{*,\Lambda} + \mathcal{O}(1)$.*
2. *For the offered wait metric, the naive policy $N(w_\ell, w_h)$, where (w_ℓ, w_h) solves (25), has the performance $w^{\Lambda}_{N(w_\ell, w_h)} = w^{*,\Lambda} + o(1)$.*

To establish an asymptotic optimality property across policies, we need to argue that a similar result holds for other feasible policies. In fact, we can extend the result in Proposition 7 to show that the performance for any policy that splits customers into $K \geq 2$ classes and servers into $K \geq 2$ pools also converges to the performance of its corresponding fluid model. Thus, because our naive two-class policy asymptotically achieves the fluid optimal solution, it is asymptotically optimal in the sense that the performance ratio of the proposed naive policy to that of any feasible policy that splits customers into multiple classes is less than one.

Table 5 shows the results for the expected queue length. We observe that the performance of the TIQ policy is indeed well approximated by the corresponding fluid limit. We observe that the accuracy gap increases as system size increases for system loads close to one (for the case $\rho = 1.05$), whereas the accuracy gap remains almost constant (independent of the system size) for high system loads (for the case $\rho = 1.5$). This is consistent with our theoretical results obtained in Proposition 7 for the naive policy. Similarly, Table 6 shows that the offered wait under the TIQ policy converges to the fluid offered wait as the system volume increases.

We next consider $M/M/n+GI$ systems operating under the TIQ policy. Although we expect the performance of the TIQ policy to asymptotically converge to

Table 6 Illustration of Asymptotic Accuracy: Expected Offered Wait Under the TIQ Policy Compared with the Corresponding Fluid Solution

	Arrival rate, Λ				
ρ	25	50	100	500	Fluid limit
Lognormal					
1.05	0.55	0.40	0.32	0.27	0.25
1.1	0.72	0.57	0.56	0.49	0.48
1.5	1.92	1.80	1.80	1.76	1.76
Erlang-3					
1.05	0.58	0.43	0.33	0.27	0.25
1.1	0.74	0.60	0.56	0.49	0.47
1.5	1.89	1.77	1.76	1.72	1.72

[4] For any real valued functions $u(\Lambda)$ and $v(\Lambda)$, we write $u(\Lambda) = \mathcal{O}(v(\Lambda))$, if $|u(\Lambda)/v(\Lambda)|$ is bounded, i.e., $\limsup_{\Lambda\to\infty} |u(\Lambda)/v(\Lambda)| < \infty$. Writing $u(\Lambda) = \mathcal{O}(1)$ simply means that $|u(\Lambda)|$ is bounded as $\Lambda \to \infty$. Finally, we write $u(\Lambda) = o(1)$ to mean that $u(\Lambda) \to 0$ as $\Lambda \to \infty$.

Table 7 Illustration of Asymptotic Accuracy: Average Fraction of Abandonment Under FCFS, Naive, and TIQ Policies Compared with the Fluid Limit

	Arrival rate, Λ				
ρ	25	50	100	500	Fluid limit
FCFS policy					
1.05	0.128	0.127	0.123	0.125	0.125
1.1	0.197	0.193	0.194	0.195	0.196
1.5	0.462	0.462	0.462	0.463	0.462
Naive policy					
1.05	0.107	0.093	0.080	0.064	0.048
1.1	0.150	0.132	0.122	0.106	0.091
1.5	0.379	0.368	0.359	0.347	0.333
TIQ policy					
1.05	0.075	0.067	0.059	0.051	0.048
1.1	0.116	0.103	0.098	0.093	0.091
1.5	0.341	0.338	0.336	0.334	0.333

the corresponding fluid limit, a formal proof requires developing additional machinery to analyze measure valued limits for $M/M/n + GI$ systems operating with non-FCFS scheduling policies. This is a worthy endeavor, but beyond the scope of our treatment. Instead, we use simulations to compute the performance under the TIQ policy and compare it with the conjectured fluid limit. In particular, we use the same study that we did in §5.2 and compare the simulated performance metrics with the corresponding values obtained from the fluid model.

5.3.2. Correlated Service and Patience Times. We next study the accuracy for the case in which service and patience times are correlated. We use the same setup as in §5.3.2 for this study. Table 7 shows the simulated average fraction of abandonment, i.e., the total abandonment rate as a fraction of the arrival rate, for each policy compared with its corresponding fluid approximation. In all cases, we observe that the fraction of abandonment converges to the fluid value. For the FCFS and TIQ policies, similar to the case of independent service and patience times, we expect the fluid approximation for the total abandonment rate to be extremely accurate. The naive policy is expected to have $\mathcal{O}(\sqrt{\Lambda})$ accuracy, and this is exhibited through the slower convergence in the table.

6. Conclusion

As customers wait in a queue, historical information can be used to gain a better understanding of how long these customers are expected to wait before abandoning the system. In this paper, we use a fluid model of the system to propose a time-in-queue policy that prioritizes customers in a dynamic fashion based on this information to optimize performance metrics of long-term average abandonment rate, queue length, and offered waits. The structure of this policy is simple and involves only two waiting time thresholds.

Our framework is general and allows customer patience times to be correlated with their service levels. For the case in which these times are independent, we provide additional insight into the optimal solution. Interestingly, we find that in this case, the three metrics we consider exhibit very different properties. For the abandonment rate metric, all nonidling policies have the same performance, and hence for this metric there is no value in differentiating customers based on time in the system. However, the queue length and offered wait metrics can be improved by differentiating customers based on the time that they have waited in the queue. In case of the queue length metric, the optimal policy utilizes the hazard rate function of the patience distribution, whereas for the offered wait metric, the optimal policy utilizes the density function of the patience distribution.

In this paper, we focused on the overloaded regime. On the fluid scale, this is the only regime that is interesting. If the system is underloaded, the fluid queue will be at a zero level, making the problem trivial. However, if the system is critically loaded, with arrival rate close to the capacity, there is potential to refine the fluid analysis by performing a diffusion analysis. This could be an interesting topic for future research.

Another point worthy of discussion is about implementation. Our proposed policies require the knowledge of the patience distribution, which requires estimation and can be a difficult task, especially when service and patience times are correlated. Combining estimating along with performance optimization is another interesting topic for future research.

Finally, we would like to point out that we have ignored the potential of strategic behavior by the customers. A natural future direction would be to consider the case where some proportion of the customers anticipates the scheduling rule of the firm and may try to behave in a manner to counter it; these customers could simply exit and re-enter the system, pretending to be new customers. Such a decision of the customer to "retry" would depend on several factors such as the time it takes to retry, their remaining patience time, value for service, cost of waiting, belief about the waiting time, etc. One approach to dealing with these strategic customers would be to formally model the retrial and patience of the customers and optimize the firm's scheduling policy. Another approach could be to counter this strategic behavior by imposing constraints on the offered waits that arise in the (nonstrategic) solution so that the difference between these is small enough to dissuade customers from retrying. We leave the analysis of both these approaches for future studies.

Appendix A. Proofs

Proof of Lemma 1. Defining $M = |\lambda(\kappa, v)|$, for each class i, $i = 1, \ldots, M$, the workload profile is given by

$$\eta_i(t) = m\lambda_i(\kappa, v)\phi(t)\mathbb{I}\{t < w_i(\kappa, v)\}. \tag{A1}$$

Thus, the overall workload profile over all classes is given by

$$\begin{aligned}\sum_{i=1}^{M}\eta_i(t) &= \sum_{i=1}^{M} m\lambda_i(\kappa, v)\phi(t)\mathbb{I}\{t < w_i(\kappa, v)\}\\ &= m\sum_{\{1\le i\le M:\, t<w_i(\kappa, v)\}}\lambda_i(\kappa, v)\phi(t)\\ &= m\Bigg(\Lambda - \sum_{\{1\le i\le|\kappa|:\, v_i\le t\}}\lambda_i(\kappa, v)\Bigg)\phi(t),\end{aligned} \tag{A2}$$

where the last equality follows from the fact that $w_i(\kappa, v) = v_i$ for $i = 1, \ldots, |\kappa|$. Noting that $\lambda_i(\kappa, v) = \kappa_i/\phi(v_i)$ for $i = 1, \ldots, |\kappa|$, it follows that $\sum_{i=1}^{M}\eta_i(t) = \eta(t)$, the workload profile under policy (κ, v) (see (2)). This completes the proof. □

Proof of Lemma 2. For any given $(\lambda, w) \in (\mathscr{L}, \mathscr{W})$ that is feasible, we will construct a feasible policy $(\kappa, v) \in (\mathscr{K}, \mathscr{V})$ with an identical workload profile. This construction is as follows: if $w_{|\lambda|} < \infty$, then we set $|\kappa| := |\lambda|$, otherwise $|\kappa| := |\lambda| - 1$; we define $v \in \mathbb{R}_+^{|\kappa|}$ as $v_i := w_i$ for $i = 1, \ldots, |\kappa|$, and $\kappa_i := m\lambda_i\phi(v_i)$ for $i = 1, \ldots, |\kappa|$. The fact that this policy has an identical workload profile follows by an argument analogous to that used in the proof of Lemma 1. Furthermore, it is easy to see that it will also have identical cost using the definition in (5). □

Proof of Proposition 1. Consider (10), and first, assume that the infimum is achieved for some $M \in \mathbb{N}$ with $(\lambda^*, w^*) \in \mathbb{R}_+^M \times \bar{\mathbb{R}}_+^M$. Then, the optimal objective value equals the solution to the following linear program:

$$\begin{aligned}\inf_{\lambda\in\mathbb{R}_+^M}\ & \sum_{i=1}^{M}\lambda_i c(w_i^*)\\ \text{s.t.}\ & \sum_{i=1}^{M}\lambda_i\phi(w_i^*) \le \frac{n}{m},\\ & \sum_{i=1}^{M}\lambda_i = \Lambda.\end{aligned}$$

Standard linear programming theory then immediately implies that there exists an optimal solution to this program that has at most two positive components. This follows because there exists an optimal solution that lies in the set of basic feasible solutions, and further, because there are only two constraints, the basis can have a rank of at most two, and thus any basic feasible solution can have at most two positive components.

We next prove that the infimum in (10) is in fact always achieved. To see this, note that for any $\epsilon > 0$, we can choose $M \in \mathbb{N}$ and $(\lambda^*, w^*) \in \mathbb{R}_+^M \times \bar{\mathbb{R}}_+^M$ such that it is within ϵ of the optimal objective value. Using the previous argument, it follows that the optimal objective value of the following program will also be within ϵ of the optimal objective value of (10):

$$\begin{aligned}\inf_{0\le\lambda_\ell,\,\lambda_h;\,0\le w_\ell<w_h\le\infty}\ & \lambda_\ell c(w_\ell) + \lambda_h c(w_h)\\ \text{s.t.}\ & \lambda_\ell\phi(w_\ell) + \lambda_h\phi(w_h) \le \frac{n}{m},\\ & \lambda_\ell + \lambda_h = \Lambda.\end{aligned}$$

However, note that the feasible set of the above optimization program is closed, and hence the optimal solution is always achieved. Furthermore, if we consider the sequence of optimizers indexed by ϵ, then as ϵ converges to zero, the corresponding objective value of this two-class optimization problem converges to the optimal objective value of (10), and hence the limit point of the two-class optimizers converges to the optimizer of (10). This completes the proof. □

Proof of Proposition 2. For any $\lambda_\ell \in [0, \Lambda]$, (w_ℓ, w_h) with $0 \le w_\ell \le w_h \le \infty$, and $\alpha > 0$, the Lagrangian corresponding to the objective function in (12) is

$$\begin{aligned}L(w_\ell, w_h, \lambda_\ell, \alpha) &= \lambda_\ell c(w_\ell) + (\Lambda - \lambda_\ell)c(w_h)\\ &\quad + \alpha(\lambda_\ell\phi(w_\ell) + (\Lambda - \lambda_\ell)\phi(w_h) - n/m).\end{aligned}$$

It follows that for optimality the following conditions are necessary:

$$c'(w_\ell) + \alpha\phi'(w_\ell) = 0, \tag{A3}$$

$$c'(w_h) + \alpha\phi'(w_h) = 0, \tag{A4}$$

$$c(w_\ell) - c(w_h) + \alpha(\phi(w_\ell) - \phi(w_h)) = 0, \tag{A5}$$

$$\lambda_\ell\phi(w_\ell) + (\Lambda - \lambda_\ell)\phi(w_h) - \frac{n}{m} = 0. \tag{A6}$$

Equations (A3), (A4), (A5), and (A6) are obtained by taking the partial derivative of $L(w_\ell, w_h, \lambda_\ell, \alpha)$ with respect to w_ℓ, w_h, λ_ℓ, and α, respectively. From (A5), we obtain

$$\frac{1}{\alpha} = -\frac{\phi(w_h) - \phi(w_\ell)}{c(w_h) - c(w_\ell)},$$

and (A3) and (A4) give us

$$\frac{1}{\alpha} = \frac{-\phi'(w_\ell)}{c'(w_\ell)} = \frac{-\phi'(w_h)}{c'(w_h)}.$$

Putting the above two relations together, we obtain (16) as the necessary conditions for optimality for solutions of the form (w_ℓ, w_h). Analogous arguments yield the necessary conditions for solutions of the form $(0, w_h)$ and (w_ℓ, ∞) given in (17) and (18). □

Proof of Proposition 3. We first prove that none of candidates for optimality listed in Proposition 2 are feasible, and thus the optimal policy would either be FCFS or LCFS.

Since r is monotone, we cannot have $r(w_\ell) = r(w_h)$; thus, (16) cannot hold. To see why (17) cannot hold, consider the setting where r is monotone decreasing. Then $r(w) > r(w_h)$ for all $w < w_h$, which implies that $-\phi'(w)/c'(w) > r(w_h)$ for all $w < w_h$, using which we have

$$\begin{aligned}\frac{\phi(w_h) - 1}{c(w_h) - c(0)} &= \frac{\int_0^{w_h}\phi'(s)\,ds}{\int_0^{w_h}c'(s)\,ds} < -r(w_h)\frac{\int_0^{w_h}c'(s)\,ds}{\int_0^{w_h}c'(s)\,ds}\\ &= -r(w_h) = \frac{\phi'(w_h)}{c'(w_h)}.\end{aligned} \tag{A7}$$

Thus, (17) cannot hold if r is monotone decreasing. A similar argument holds if r is monotone increasing. Similar arguments can be used to obtain that (18) cannot hold as well.

Next, we compare FCFS and LCFS. To do so, we consider policies of the form $(0, w)$ for $\bar{w} \le w \le \infty$ and note that the cases $w = \bar{w}$ and $w = \infty$ correspond to FCFS and LCFS, respectively. For such policies, we denote the objective function in (12) by $\Pi(w)$ and write it out using (14) as follows:

$$\Pi(w) = \Lambda c(w) + \frac{n - m\Lambda\phi(w)}{m(1-\phi(w))}(c(0) - c(w)). \tag{A8}$$

Then, differentiating with respect to w, we obtain

$$\begin{aligned}\Pi'(w) &= \Lambda c'(w) - \frac{n - m\Lambda\phi(w)}{m(1-\phi(w))}c'(w) - \frac{m\Lambda\phi'(w)}{m(1-\phi(w))} \\ &\quad \cdot (c(0) - c(w)) + \phi'(w)\frac{n - m\Lambda\phi(w)}{m(1-\phi(w))^2}(c(0) - c(w)) \\ &= \frac{m\Lambda - n}{m(1-\phi(w))}\left(c'(w) - \phi'(w)\frac{c(0) - c(w)}{1-\phi(w)}\right) \\ &= c'(w)(c(w) - c(0))\frac{m\Lambda - n}{m(1-\phi(w))^2} \\ &\quad \cdot \left(\frac{1-\phi(w)}{c(w) - c(0)} - r(w)\right). \end{aligned} \tag{A9}$$

Thus, noting that $c'(w) > 0$, $c(w) > c(0)$, $m\Lambda > n$, and there is no $w \in (0, \infty)$ that satisfies $r(w) = (1-\phi(w))/(c(w) - c(0))$, it follows that $\Pi'(w)$ is either always positive or negative. If r is monotone decreasing, then relation (A7) holds for every $w_h = w$, i.e., we have $r(w) < (1-\phi(w))/(c(w) - c(0))$ and we obtain that $\Pi'(w) > 0$ for all $w > 0$. Thus, $\Pi(w)$ is minimized at the smallest w possible, $w = \bar{w}$, and we obtain that FCFS is optimal. Similarly, if r is monotone increasing, then we obtain $\Pi'(w) < 0$ so that the objective is minimized at the largest w possible, which is ∞, and we obtain that $(0, \infty)$ or LCFS is optimal. □

Proof of Proposition 4. The proof follows immediately by noting that for all policies (w_ℓ, w_h) with associated arrival rates $(\lambda_\ell, \lambda_h)$ that satisfy constraints (21)–(22), the objective function value is a constant and equals $\Lambda - n/m$. □

Proof of Propositions 5 and 6. Both these results follow immediately from the following general result.

Lemma 3. *If r is quasi-concave and nonmonotone, then we have the following*:

1. *If $\lim_{w\to\infty} r(w)(c(w) - c(0)) \ge 1$, then LCFS is optimal.*
2. *If $\lim_{w\to\infty} r(w)(c(w) - c(0)) < 1$, then there is a unique solution $w_h \in (0, \infty)$ to (17). If $w_h > \bar{w}$, then $(0, w_h)$ is the optimal solution to (15); otherwise FCFS is optimal.*

Proof. We begin by noting that if r is quasi-concave, then the policies $0 < w_\ell < \bar{w} < w_h < \infty$ identified in Proposition 2.1 cannot be optimal because the first-order condition (16) cannot be satisfied. To see this, consider any (w_ℓ, w_h) such that $r(w_\ell) = r(w_h)$; then $r(w) > r(w_\ell)$ for all $w_\ell < w < w_h$. The latter implies that $-\phi'(w)/c'(w) > r(w_\ell)$ for all $w_\ell < w < w_h$, using which we have

$$\begin{aligned}\frac{\phi(w_h) - \phi(w_l)}{c(w_h) - c(w_l)} &= \frac{\int_{w_\ell}^{w_h}\phi'(s)\,ds}{\int_{w_\ell}^{w_h}c'(s)\,ds} < -r(w_\ell)\frac{\int_{w_\ell}^{w_h}c'(s)\,ds}{\int_{w_\ell}^{w_h}c'(s)\,ds} \\ &< -r(w_\ell) = \frac{\phi(w_\ell)}{c(w_\ell)}. \end{aligned} \tag{A10}$$

Thus, (16) cannot hold.

Next, we consider policies of the form (w, ∞) for $0 \le w \le \bar{w}$. We only consider the case in which $c(\infty) < \infty$; otherwise, we can disregard these policies. We will prove that of such policies, only FCFS and LCFS can be optimal. We denote the objective function in (12) by $\Pi(w)$ and write it out using (14), which gives us $\lambda_\ell(w, \infty) = n/m\phi(w)$ so that we can write

$$\Pi(w) = \Lambda c(\infty) - \frac{n}{m\phi(w)}(c(\infty) - c(w)). \tag{A11}$$

Then, differentiating with respect to w, we obtain

$$\begin{aligned}\Pi'(w) &= \frac{n}{m\phi(w)}c'(w) + \frac{n\phi'(w)}{m\phi(w)^2}(c(\infty) - c(w)) \\ &= c'(w)(c(\infty) - c(w))\frac{n}{m\phi(w)^2}\left(\frac{\phi(w)}{c(\infty) - c(w)} - r(w)\right). \end{aligned}$$

Suppose $r(w)$ is increasing on $[0, w_r)$ and decreasing on (w_r, ∞). We first consider $w \ge w_r$. Since r is decreasing on (w_r, ∞), we have $r(w) < r(w_r)$ for all $w > w_r$. This implies that $-\phi'(w)/c'(w) < r(w_r)$ for all $w > w_r$, using which we have

$$\frac{-\phi(w)}{c(\infty) - c(w)} = \frac{\int_w^\infty \phi'(s)\,ds}{\int_w^\infty c'(s)\,ds} > -r(w)\frac{\int_w^\infty c'(s)\,ds}{\int_w^\infty c'(s)\,ds} = -r(w). \tag{A12}$$

Then, $\Pi'(w) < 0$ for $w \ge w_r$.

Suppose $\Pi'(w) = 0$ has no solution; then it follows that the optimal policy of the form (w, ∞) would involve choosing the largest w possible, which is $\bar{w}$, and hence FCFS would be optimal. Suppose there exists $\tilde{w}$ such $\Pi'(\tilde{w}) = 0$. We next prove that $\tilde{w}$ must be a local maximizer. To see this, we compute

$$\begin{aligned}\Pi''(\tilde{w}) &= c'(\tilde{w})(c(\infty) - c(\tilde{w}))\frac{n}{m\phi(\tilde{w})^2} \\ &\quad \cdot \left(\frac{(c(\infty) - c(\tilde{w}))\phi'(\tilde{w}) + c'(\tilde{w})\phi(\tilde{w})}{(c(\infty) - c(\tilde{w}))^2} - r'(\tilde{w})\right) \\ &= -c'(\tilde{w})(c(\infty) - c(\tilde{w}))\frac{n}{m\phi(\tilde{w})^2}r'(\tilde{w}) < 0, \end{aligned} \tag{A13}$$

where we use the fact that $\tilde{w} < w_r$, which implies that $r'(\tilde{w}) > 0$. Thus, $\tilde{w}$ is a local maximizer, and we obtain that among policies of the form (w, ∞), we only need to consider FCFS and LCFS (so that the w is largest or smallest possible). Noting that these policies are also obtained when considering policies of the form $(0, w)$ for $\bar{w} \le w$, we now restrict attention to analyzing policies of form $(0, w)$ without any loss of generality.

For policies of the form $(0, w)$, we proceed as in the proof of Proposition 3 and can compute $\Pi'(w)$ as in (A9). Arguing as in that proof, we obtain that $\Pi'(w) < 0$ for $0 < w \le w_r$.

Suppose there exists $\tilde{w}$ such that $\Pi'(\tilde{w}) = 0$. Then, analogous to (A13), we can compute the second derivative of the objective function to obtain

$$\begin{aligned}\Pi''(\tilde{w}) &= c'(\tilde{w})(c(\tilde{w}) - c(0))\frac{m\Lambda - n}{m(1-\phi(\tilde{w}))^2} \\ &\quad \cdot \left(\frac{-(c(\tilde{w}) - c(0))\phi'(\tilde{w}) - c'(\tilde{w})(1-\phi(\tilde{w}))}{(c(\tilde{w}) - c(0))^2} - r'(\tilde{w})\right) \\ &= -c'(\tilde{w})(c(\tilde{w}) - c(0))\frac{m\Lambda - n}{m(1-\phi(\tilde{w}))^2} r'(\tilde{w}) > 0,\end{aligned}$$

where we use the fact that $\tilde{w} > w_r$ so that $r'(\tilde{w}) < 0$. It follows that $\tilde{w}$ is the unique minimizer of $\Pi(w)$ and further that it is the only extrema point. So, we obtain that $\Pi'(w) = 0$ can have at most one solution on $(0, \infty)$, and if such a solution $\tilde{w}$ exists, then it is the global minimizer of $\Pi(w)$. So, noting that a policy $(0, \tilde{w})$ is feasible only for $\tilde{w} > \bar{w}$, we obtain that if $\Pi'(w) = 0$ has a solution $\tilde{w}$, then if $\tilde{w} > \bar{w}$, the optimal policy is $(0, \tilde{w})$; otherwise, because $\Pi(w)$ is increasing for $w > \tilde{w}$, the optimal policy is FCFS. If $\Pi'(w) = 0$ has no solution, then $\Pi'(w) < 0$ for all w, and the optimal solution is $(0, \infty)$ or LCFS.

Finally, we derive the necessary and sufficient condition for $\Pi'(w) = 0$ to have an interior solution. To do so, we write $\Pi'(w)$ (given in (A9)) as follows:

$$\Pi'(w) = c'(w)\frac{m\Lambda - n}{m(1-\phi(w))^2}(1 - \phi(w) - r(w)(c(w) - c(0))).$$

Defining $g(w) := (1 - \phi(w) - r(w)(c(w) - c(0)))$ and noting that $c'(w) > 0$, we obtain that $\Pi'(w) = 0$ for some $w \in (0, \infty)$ is equivalent to $g(w) = 0$. Furthermore, we have $g'(w) = -r'(w)(c(w) - c(0))$, and thus g is unimodal with a minima at $w = w_r > 0$. Noting that $g(0) = 0$, it follows that $g(w) = 0$ has a solution $w \in (0, \infty)$ if and only if $\lim_{w\to\infty} g(w) > 0$, which is equivalent to $\lim_{w\to\infty} r(w)(c(w) - c(0)) < 1$. This completes the proof. □

Proof of Proposition 7. The result follows by establishing an asymptotic result for each class separately and then combining it. The class with offered wait $w_h > 0$ is overloaded, and we can apply the results in Bassamboo and Randhawa (2010) to obtain that the expected queue length for this class is within $\mathcal{O}(1)$ of the fluid approximation. Turning to the other class, if $w_\ell > 0$, then we obtain a similar result for this class, and combining the queue length, we obtain that the overall queue length is within $\mathcal{O}(1)$ of the fluid approximation. If $w_\ell = 0$, then that class is critically loaded, so its fluid approximation is zero, and we can use the results in Zeltyn and Mandelbaum (2005) to obtain that its queue length is $\mathcal{O}(\sqrt{\Lambda})$. This completes the proof of part 1. For part 2, we directly apply the results in Zeltyn and Mandelbaum (2005) to obtain that the offered wait under the naive policy for each class asymptotically converges to its fluid approximation for each class. □

Appendix B. Optimal Policies for Dependent Service and Patience Times

Section 4.1 characterized the optimal policy using the function r, which is the ratio of the decrease in remaining work to the increase in cost as a function of waiting time. Here, we define a function that captures the remaining conditional workload and show how properties of this function can be used to obtain more insight into the optimal policies. We define the remaining conditional workload function as

$$z(w) := m\frac{\phi(w)}{\bar{F}_A(w)}. \tag{B1}$$

The function $z(w)$ measures the amount of work remaining per remaining customer of those who arrived w time units earlier.

Proposition 8. 1. *For the abandonment criterion, i.e., $c(w) = F_A(w)$, if the marginal patience distribution has a nondecreasing hazard rate and $z(w)$ is concave increasing (convex decreasing), then the optimal policy is LCFS (FCFS).*

2. *For the queue length criterion, i.e., $c(w) = \int_0^w \bar{F}_A(x)\,dx$, if $z(w)$ is concave increasing (convex decreasing) and the marginal patience distribution has a nondecreasing (nonincreasing) hazard rate, then the optimal policy is LCFS (FCFS).*

3. *For the offered wait criterion, i.e., $c(w) = w$, if $z(w)$ is convex decreasing and the (marginal) patience distribution has a nonincreasing density, then the optimal policy is FCFS. Furthermore, LCFS is never an optimal policy.*

The proof follows by using Proposition 3 and straightforward calculations, and is omitted for brevity.

References

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

Atar R, Giat C, Shimkin N (2010) The $c\mu/\theta$ rule for many-server queues with abandonment. *Oper. Res.* 58(5):1427–1439.

Atar R, Kaspi H, Shimkin N (2014) Fluid limits for many-server systems with reneging under a priority policy. *Math. Oper. Res.* 39(3):672–696.

Bassamboo A, Randhawa RS (2010) On the accuracy of fluid models for capacity planning in queueing systems with impatient customers. *Oper. Res.* 58(5):1398–1413.

Bassamboo A, Harrison JM, Zeevi A (2006) Design and control of a large call center: Asymptotic analysis of an LP-based method. *Oper. Res.* 54(3):419–435.

Bassamboo A, Randhawa RS, Zeevi A (2010) Capacity sizing under parameter uncertainty: Safety staffing principles revisited. *Management Sci.* 56(10):1668–1686.

Dai J, Tezcan T (2008) Optimal control of parallel server systems with many servers in heavy traffic. *Queueing Systems* 59(2): 95–134.

Down DG, Koole G, Lewis ME (2011) Dynamic control of a single-server system with abandonments. *Queueing Systems* 67(1): 63–90.

Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. *Manufacturing Service Oper. Management* 5(2):79–141.

Ghamami S, Ward AR (2013) Dynamic scheduling of a two-server parallel server system with complete resource pooling and reneging in heavy traffic: Asymptotic optimality of a two-threshold policy. *Math. Oper. Res.* 38(4):761–824.

Gurvich I, Whitt W (2010) Service-level differentiation in many-server service systems via queue-ratio routing. *Oper. Res.* 58(2): 316–328.

Jouini O, Koole G, Roubos A (2013) Performance indicators for call centers with impatience. *IIE Trans.* 45(3):341–354.

Kang W, Ramanan K (2010) Fluid limits of many-server queues with reneging. *Ann. Appl. Probab.* 20(6):2204–2260.

Mandelbaum A, Zeltyn S (2013) Data stories about (im)patient customers in tele-queues. *Queueing Systems* 73(4):1–32.

Pinedo ML (2012) *Scheduling: Theory, Algorithms, and Systems* (Springer, New York).

Reich M (2012) The offered-load process: Modeling, inference and applications. Master's thesis, Department of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa.

Whitt W (2006) Fluid models for multiserver queues with abandonments. *Oper. Res.* 54(1):37–54.

Zeltyn S, Mandelbaum A (2005) Call centers with impatient customers: Many-server asymptotics of the $M/M/n + G$ queue. *Queueing Systems* 51(3-4):361–402.

Zhang J (2013) Fluid models of many-server queues with abandonment. *Queueing Systems* 73(2):147–193.