Dynamic Scheduling in a Many-Server, Multiclass System: The Role of Customer Impatience in Large Systems

Jeunghyun Kim, Ramandeep S. Randhawa, Amy R. Ward

To cite this article:

Full terms and conditions of use: http://pubsonline.informs.org/page/terms-and-conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article’s accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2018, INFORMS
Dynamic Scheduling in a Many-Server, Multiclass System: The Role of Customer Impatience in Large Systems

Jeunghyun Kim, Ramandeep S. Randhawa, Amy R. Ward

Abstract. Problem definition: We study optimal scheduling of customers in service systems, such as call centers. In such systems, customers typically hang up and abandon the system if their wait for service is too long. Such abandonments are detrimental for the system, and so managers typically use scheduling as a tool to mitigate it. In this paper, we study the interplay between customer impatience and scheduling decisions when managing heterogeneous customer classes. Academic/practical relevance: Call centers constitute a large industry that has a global spending of around $300 billion and employs more than 15 million people worldwide. Our work focuses on improving call center operations, which can reduce costs and improve customer satisfaction. Mathematically, customer patience is typically modeled as exponentially distributed for tractability. Our work makes inroads into relaxing this restrictive assumption to allow modeling more realistic call center situations. Methodology: We use heavy traffic–motivated asymptotic queueing machinery that provides us the traction to successfully capture and incorporate the customer impatience distribution into the scheduling problem. In our approach, the scheduling problem reduces to a diffusion control problem, which we solve to propose near-optimal scheduling policies. Results: We propose near-optimal scheduling policies that can be implemented by call centers to improve their quality-of-service metrics. One of our main results is that, for a class of parameters, we establish sufficient conditions for both the optimality and nonoptimality of threshold policies. Managerial implications: Threshold policies are widely used for scheduling. Our work provides additional insight into whether these may be suboptimal. Our work provides an easy-to-implement alternative that can reduce customer abandonments considerably; for instance, our numerical results indicate that for a system with two customer types, the abandonment rate of one class can be lowered by 30% by using our policy relative to the best threshold policy.

1. Introduction

The study of scheduling problems has a long history in the academic literature (Pinedo 1995). One celebrated result within the context of stochastic scheduling problems is the $c \mu$ rule, which is provably optimal (in an exact or asymptotic sense) under a wide variety of assumptions (Smith 1956, Van Mieghem 1995, Mandelbaum and Stolyar 2004). However, that rule often fails in the presence of time-sensitive customers (Rubino and Ata 2009, Down et al. 2011), who are common in service systems. Our objective in this paper is to identify simple and effective rules to replace $c \mu$ in such settings.

Statistically, an important observation about the willingness to wait for service of a time-sensitive customer, termed the customer’s patience time, is that it changes over time (Brown et al. 2005, Mandelbaum and Zeltyn 2013). In particular, as customers wait for their service to begin, some become more patient (for example, as a result of having already paid a waiting cost), and some become less patient (for example, as a result of being frustrated from waiting). The novelty in this work is to devise scheduling rules that account for this temporal change in customer patience. Technically, this means not assuming patience times follow an exponential distribution, which is a very common assumption in the literature.

We modify a well-studied model for service systems with homogeneous customers, the $M/M/N+GI$ queue (Zeltyn and Mandelbaum 2005), to account for heterogeneous customers. Our multiclass $M/M/N+GI$ queue distinguishes different customer classes based on their patience distribution, their arrival rate, and the cost incurred if a customer abandons (that is, leaves without receiving service). The scheduling decision is how to pair a newly available server with a customer...
when there are customers from more than one class waiting to minimize long-run average cost. Note that when the costs associated with each customer class are the same, the objective of minimizing long-run average cost exactly coincides with maximizing the throughput rate.

The scheduling decision is, in general, dynamic because it can depend on the relative number of customers from the different classes. The optimization problem to determine optimal scheduling is very complicated and not amenable to exact analysis. This leads us to apply fairly standard approximation techniques used in the extant literature. In particular, we assume that the service system has high customer demand, a large number of servers, and is operating critically loaded. Formally, we consider the Halfin–Whitt many-server scaling in the quality- and efficiency-driven regime. In this asymptotic regime, our scheduling problem reduces to a diffusion control problem (DCP). It is sufficient to study the solution to the DCP because those solutions readily translate to scheduling policies for a $M/M/N+GI$ queue.

A summary of our main results is as follows. We analytically solve the underlying DCP to propose a scheduling policy for a multiclass $M/M/N+GI$ queue with multiple impatient customer types. We analyze the structure of the policy and identify sufficient conditions when this policy is a static control or threshold policy. We also find conditions under which static control and threshold policies are suboptimal. In such cases, we find that our proposed policy can have an interesting structure (such as the U-shape described subsequently), and further, our numerical results show that our proposed policy can lower customer abandonments by as much as 30% relative to the best static control or threshold policies. Our numerical results also show that the proposed policy is robust to the estimation of system parameters and the regime in which the system operates.

Formally, when we analyze the underlying diffusion control problem, we find that there arise two potential approximating problems: one in which the underlying diffusion has linear drift and is based on the value of the patience-time density at zero and one in which the underlying diffusion has nonlinear drift and incorporates the entire patience-time density function. We solve the DCPs by identifying the Hamilton–Jacobi–Bellman (HJB) equations that characterize an optimal scheduling policy. We show that there exists a unique solution to the HJB equations, implying that an optimal scheduling policy can be computed numerically. We view the HJB equation as a map that connects the patience distribution and the optimal scheduling policy, and we use this map to derive results in the form of an optimal policy. This leads to the following main contributions of this paper.

1. **Linear drift DCP.** We show that the DCP solution either leads to a static class ranking or is of threshold form. The key to identifying the optimum thresholds is to first understand which classes will have priority that change in accordance with the system state and which classes will have a higher priority always. We provide a simple algorithm to do this.

2. **General DCP.** We show that in a nontrivial portion of the parameter space, the DCP solution does not either lead to a static class ranking or have a threshold form. For a two-class model, we provide conditions on the patience distributions under which (i) the aforementioned forms emerge or (ii) the optimum policy has a different structure.

For the general DCP, one novel structure we see emerge from our proposed policy is a U-shape as shown in Figure 1, which has rich intuition. Consider a two-class model in which class 1 has a patience-time distribution with increasing hazard rate, class 2 has an exponential patience-time distribution, and class 1 has a lower cost if a customer’s patience runs out before that customer’s service begins. When there are few customers in the system, allowing the cheaper class 1 to leave without receiving service is beneficial. As the system becomes more congested, the increasing class 1 hazard rate is evidenced through more and more class 1 customers leaving, resulting in increased cost. That effect is counteracted by giving some priority to the class 1 customers. Eventually, when the system is very congested, the forward-looking system manager returns to fully prioritizing class 2 customers, using the very high abandonment rate of the class 1 customers to “trim” the long-run average system congestion level. To give a better perspective, Figure 1 also shows a threshold structure (the dashed line, blue in the online version).

The remainder of this paper is organized as follows. We end the introduction by reviewing the related

![Figure 1. (Color online) Description of the Proposed Policy and Threshold Policy](image-url)
literature. Section 2 sets up the model and formulates the optimization problem. Section 3 summarizes our main results. Section 4 formulates the relevant DCP, sets up the HJB equations, and proves the existence of a solution to the HJB equations that solves the DCP. We show that the solution to the linear HJB equations either gives a static class ranking or is of threshold form in Section 5 but that this result does not extend to the general HJB equations. For a two-class model, Section 6 provides conditions on the patience distributions for when the solution to the HJB equations is either a static ranking or has a threshold form and further shows that the solution has a U-shape when those conditions are not satisfied. Finally, we make concluding remarks in Section 7. All the proofs of technical results can be found in the electronic companion to this paper; we also discuss how our results can be extended to include heterogeneous service rates in the electronic companion.

1.1. Literature Review

Our study relates to the literature on scheduling in queuing systems. The most widely known result from this literature is the optimality of policies (including the static priority policy known as the $c\mu$ rule) that greedily prioritizes customers in the order of customers’ instantaneous costs. Recently, Atar et al. (2010, 2011) extended the optimality of the $c\mu$ rule to a system with impatient customers modeled as an overloaded multiclass $M/M/N+M$ system. However, the optimality of greedy policies in the presence of customer abandonment does not extend in general. Instead, scheduling policies that dynamically prioritize customers have been proposed (see Atar et al. 2004, Harrison and Zeevi 2004, Gurvich and Whitt 2009, Rubino and Ata 2009, and Ghamami and Ward 2013 for critically loaded systems; see Down et al. 2011 for lightly loaded ones).

The aforementioned papers all assume that the patience-time distributions are exponential. For reasons of mathematical tractability, assuming exponential patience time is a common practice in papers that aim to provide useful insights on managing service operations (Gurvich et al. 2008, Gurvich and Whitt 2010). Our paper differs from these papers as we use generally distributed patience time. The departure from the exponential patience-time assumption is motivated by evidence found in empirical or experimental research; see Brown et al. (2005) and Mandelbaum and Zeltyn (2013) for empirical evidence and Kort (1983) for experimental evidence.

The papers that consider general patience-time distributions are recent, and there are two streams of works therein. The first stream builds machinery to approximate systems with general patience-time distributions. Whitt (2006) proposes the fluid approximation of an overloaded single class $G/GI/N+GI$ system. Kang and Ramanan (2010) and Zhang (2013) prove the convergence to the fluid model proposed in Whitt (2006) under different assumptions on service-time distribution. Atar et al. (2013) considers an overloaded multiclass $G/GI/N+GI$ system operated under the $c\mu$ rule and proves the accuracy of the fluid approximation. In that paper, when the patience-time distribution is exponential, the $c\mu$ rule is proven to be asymptotically optimal. Finally, Bassamboo and Randhawa (2010) study the accuracy of fluid models for a single-class $M/M/N+GI$ system and solves a capacity sizing problem. The authors show that the optimal operating regime differs based on the structure of the hazard rate function of the patience-time distribution.

Moving from fluid to diffusion approximations, Ward and Glynn (2005), Dai and He (2010, 2011), and Mandelbaum and Momčilović (2012) use the value of the patience-time density function at the origin to approximate customer abandonments and prove its accuracy. Other papers use the entire distribution to approximate customer abandonments. Reed and Ward (2008) introduce a diffusion approximation of a critically loaded single-class $GI/GI/1+GI$ system based on the hazard rate function of the patience-time distribution and prove the accuracy of the approximation. Reed and Tezcan (2012) prove a similar hazard rate approximation for a critically loaded single-class $GI/M/N+GI$ system. Dai and He (2013) propose a hazard rate–based diffusion model for a critically loaded single-class $GI/Ph/N+GI$ system and devise an efficient numerical algorithm to calculate its steady-state distribution. Weerasinghe (2014) shows that the hazard rate approximation for a critically loaded single-class $G/M/N+GI$ system with state-dependent service rates is accurate. Katsuda (2015) relaxes some regularity conditions imposed on the patience-time distribution in the literature and proves the accuracy of the hazard approximation for a critically loaded single-class $G/Ph/N+GI$ system. Huang et al. (2016) propose a unifying approximation scheme that includes the ones based on the patience-time density function considered only at the origin and the ones that consider the entire hazard rate and show that this unified approximation is accurate for a critically loaded single-class $G/GI/N+GI$ system.

The second stream of the papers that consider general patience-time distributions utilizes the machinery developed in the first stream to solve optimization problems for systems with general patience-time distributions. In the scheduling literature, there are three such papers. Considering an overloaded single-class $G/GI/N+GI$ system, Bassamboo and Randhawa (2015) show the benefit of prioritizing a priori homogeneous customers based on their actual waiting times instead of serving them in an FCFS manner. Our work differs because we optimize scheduling decisions
across different classes (distinguished by patience-time distributions and associated abandonment costs) but assume customers are served in the FCFS manner within the same class. Long and Zhang (2015) propose a scheduling policy for a multiclass $G/G/N+G$ system and prove its asymptotic optimality in the fluid scale when the patience-time distribution for each class has a decreasing hazard rate. In our work, the hazard rate function can take any form, and we propose a scheduling policy based on the optimization problem in the diffusion scale. Kim and Ward (2013) solve a similar problem to the one considered here for a single-server setting by looking at a critically loaded multiclass $GI/GI/1+GI$. Our paper is different from Kim and Ward (2013) in that we study a many-server system $(M/M/N+GI)$ and relax the assumed increasing hazard rate of the patience time from Kim and Ward (2013). The main intuition that allows that assumption to be dropped is that this paper only focuses on stationary controls when we solve the approximating DCP whereas Kim and Ward (2013) allow for a larger class of admissible policies.

In summary, we find no policy that bears any resemblance to the U-shape structure in Figure 1 in any of the aforementioned papers.

2. Model

We model the service system as an $M/M/N+GI$ queue. Customers arrive to this system as a Poisson process with rate $\lambda$. The work amount from each customer is exponentially distributed with mean $1/\mu$, and servers are homogeneous in the sense that they all work at rate one. Also, servers are fully flexible; that is, every server can serve any customer. Each customer has a randomly distributed patience time and abandons the system without being served if the service is not commenced within the patience time.

Different customers exhibit different impatience behaviors. For example, after having waited a while, some will become less sensitive to the delay and become less likely to abandon. Others will become increasingly frustrated as their wait time increases and become more likely to abandon. We want to model these different reactions of customers with regard to waiting. In particular, we assume that there are $K \geq 2$ different types of patience-time distributions, $G_k$ for $k \in \mathcal{K} := \{1, \ldots, K\}$, and a class $k$ customer’s patience time is drawn from $G_k$. We assume that the interior of the domain of $G_k$ is $(0,d_k)$ for some $d_k \leq \infty$ and $G_k$ has a well-defined density function $g_k$. The rate at which class $k$ customers arrive is given by $a_k \lambda$ for some positive constant $a_k$ such that $\sum_{k \in \mathcal{K}} a_k = 1$. All random variables are assumed to be independent of each other.

When customers are impatient, customer abandons indicate customers’ dissatisfaction toward the system. Consequently, each abandonment potentially imposes a cost to the system, and the system manager wants to minimize this cost associated with customer abandonments. Given the classification of customers, a relevant decision lever is the scheduling policy that describes which customer an available server will serve next when there are customers from multiple classes waiting. Our goal is to devise a scheduling policy that minimizes the abandonment cost. One intuition to minimize the abandonment cost is to always prioritize the class with the highest instantaneous abandonment cost rate. However, such a policy is myopic. If, for example, each abandonment from that class is relatively cheap, then the system manager may prefer to let that class abandon and to use those abandonments to trim the overall system congestion, thereby incurring short-term pain in exchange for long-term benefit. To formally study the design of an optimal scheduling policy, we assume there is a class $k$ abandonment penalty $r_k$ and find a scheduling policy $\pi$ that minimizes the long-run average cost:

$$c(\pi) := \limsup_{T \to \infty} \mathbb{E} \left[ \sum_{k \in \mathcal{K}} r_k (T; \pi_k) \right].$$

In (1), $R_k(T; \pi)$ denotes the cumulative number of class $k$ customers that have abandoned in $[0,T]$ under the scheduling policy $\pi$. Note that, when $r_k = r$ for some constant $r > 0$ and all $k \in \mathcal{K}$, minimizing (1) is equivalent to maximizing the throughput rate of the system. In a revenue-generating system, the penalty $r_k$ can be interpreted as the revenue lost when a class $k$ customer abandons.

The class of scheduling policies we consider are those that (i) do not assume knowledge of the future, (ii) enforce that once a customer enters service that customer stays in service until completion, (iii) do not allow servers to idle while customers are waiting, and (iv) are stationary. Mathematically, the class of admissible scheduling policies $\Pi$ consists of all nonanticipating, nonpreemptive, and nonidling policies that are stationary. We would like to find $\pi^* \in \Pi$ such that the long-run average cost limit exists and attains the minimum possible cost; that is,

$$c(\pi^*) = \lim_{T \to \infty} \mathbb{E} \left[ \sum_{k \in \mathcal{K}} r_k (T; \pi^*) \right] = \inf_{\pi \in \Pi} c(\pi).$$

In general, solving (2) is very complicated. This is because writing the state evolution equations must involve measure-valued processes that track the time until each customer’s patience is exhausted; see, for example, Kang and Ramanan (2010) and Zhang (2013). The implication is that calculating the long-run average abandonment rate for any fixed scheduling policy is mathematically not tractable. Our approach is to identify a limit regime under which we expect the solution to (2) to be close to the solution to an approximating...
diffusion control problem (DCP). Then, we solve the DCP, interpret that solution as an admissible policy, and validate its performance numerically. The analytic tractability of the DCP allows us to classify its solution in terms of properties of the patience time distributions, from which follows a connection between the structure of a near-optimal policy and the patience-time distribution.

We consider large firms with many servers and much demand. In general, there will be too much capacity, too little capacity, or the capacity will approximately balance supply and demand. Our focus is the case in which the capacity approximately balances supply and demand. Specifically, we assume

\[ N = \frac{\lambda}{\mu} + \beta \sqrt{\frac{\lambda}{\mu}} \]  

(3)

for some constant \( \beta \in \mathbb{R} \), so that the system operates in the so-called quality- and efficiency-driven (QED) regime (Halfin and Whitt 1981, Garnett et al. 2002). Still, by varying \( \beta \) in (3), our study can provide some understanding of optimal scheduling decisions in overloaded and underloaded regimes as well (see Section 3).

In the QED regime, the main challenge is to manage the stochastic fluctuations created by the variability in the arrival processes and service times. In the underloaded regime, in which there is too much capacity in the sense that \( N = (1 + \epsilon) (\lambda / \mu) \) for some constant \( \epsilon > 0 \) independent of \( \lambda \), the scheduling policy becomes irrelevant because there is very little abandonment under any scheduling policy. On the other hand, in the overloaded regime, in which there is too little capacity in the sense that \( N = (1 - \epsilon) (\lambda / \mu) \) for some constant \( \epsilon > 0 \) independent of \( \lambda \), the main challenge to address comes from the imbalance between demand and service capacity. This case makes fluid approximations, such as the one in Atar et al. (2010), relevant. The main message of Atar et al. (2010) is that a static priority policy is optimal with too little capacity. However, as pointed out earlier in the literature review, in the QED regime, static priority has been shown to be optimal in more generality.

2.1. Notation

For any \( a \in \mathbb{R} \), we denote \( \max\{a, 0\} \) and \( -\min\{a, 0\} \) by \( a^+ \) and \( a^- \), respectively. Also, for an increasing function \( h \), we use \( h^{-1} \) to denote its inverse; that is, \( h^{-1}(y) := \sup\{x \geq 0 : h(x) \leq y\} \) for \( y \geq 0 \) with the convention that \( h^{-1}(y) = 0 \) for all \( y < h(0) \) and \( h^{-1}(y) = \infty \) for all \( y > \sup_{x \geq 0} h(x) \).

3. Summary of Main Results

3.1. Proposing a Scheduling Policy

We first derive and next solve the approximating DCP of the scheduling problem to obtain a function \( q^*: \mathbb{R} \rightarrow [0, 1]^K \) such that \( \sum_{k \in \mathbb{R}} q^*_k(x) = 1 \) for all \( x \geq 0 \) (see Section 4). Our policy is nonpreemptive and attempts to maintain the proportion of waiting customers at \( q^* \).

We do so by allocating a newly idle server to process a customer from the class that exceeds the most from the amount given by \( q^* \). Formally, define \( Q_k(t; \pi) \) as the number of class \( k \) customers waiting for the service at time \( t \) under a scheduling policy \( \pi \in \Pi \). Under our proposed policy, an available server at time \( t \) serves a customer at the head of the class \( i \) queue where

\[ i \in \arg\max_{k \in \mathbb{R}} \left\{ \frac{Q_k(t; \pi)}{\sum_{j \in \mathbb{R}} Q_j(t; \pi)} - q^*_k \left( \sum_{j \in \mathbb{R}} Q_j(t; \pi) \right) \right\}. \]

(4)

The function \( q^* \) is defined by finding the function \( v^*: \mathbb{R} \rightarrow \mathbb{R} \) and constant \( \kappa < \infty \) that solve the ordinary differential equation (ODE)

\[ \lambda v'(x) + (\sqrt{\mu} x - \beta \sqrt{\lambda}) \sqrt{\mu v(x)} + \lambda \min_{q \in \mathcal{Q}} \phi(x, v(x), q) = \kappa, \]

such that \( \sup_{x < \alpha} |v(x)| < \infty \),

(5)

where \( d := \lambda \sum_{k \in \mathbb{R}} a_k d_k \) represents the maximum possible number of customers waiting,

\[ \phi(x, w, q) := \sum_{k \in \mathbb{R}} (r_k - w) m_k \left( \frac{x^+ q_k}{\lambda} \right) \]

for \( x \leq d, w \in \mathbb{R}, \) and \( q \in \mathcal{Q} := ((q_1, \ldots, q_K): \sum_{k \in \mathbb{R}} q_k = 1, q_k \geq 0 \) for \( k = 1, \ldots, K \), and

\[ m_k(x) := a_k \int_{0}^{x/a_k} h_k(y) \, dy, \]

for \( x \geq 0 \),

(6)

where \( h_k \) denotes the hazard rate function associated with the class \( k \) patience time distribution. Then,

\[ q^*(x) = \arg\min_{q \in \mathcal{Q}} \left\{ \sum_{k \in \mathbb{R}} r_k m_k \left( \frac{x^+ q_k}{\lambda} \right) - \sum_{k \in \mathbb{R}} v^*(x) m_k \left( \frac{x^+ q_k}{\lambda} \right) \right\} \]

(7)

(see Theorem 1). The minimum exists because the domain of \( \mathcal{Q} \) is compact and the functions \( m_k, k \in \mathbb{R}, \) and \( v^* \) are continuous. Although we cannot provide a general structure for \( q^* \), we can always solve for \( q^* \) numerically. When there are two classes, we find that nondecreasing hazard rates often leads to the U-shape structure depicted in Figure 1.

3.2. Both Static Priority and Threshold Controls May Be Suboptimal

We establish that static priority and threshold controls are not optimal in general. We find sufficient conditions both for their optimality and suboptimality (see Propositions 1–5 in Sections 5 and 6). We further find that when these controls are suboptimal, there can be large
benefits to using our proposed policy. We illustrate this numerically using a two customer–type setting. In such a system, a threshold control is one that prioritizes one class for system workloads below a threshold and the other class for system workloads that exceed the threshold; a static priority control is essentially a threshold control with a threshold that equals zero or infinity.

We compare the performance of the best threshold policy (found by a brute force search on the threshold level) and our proposed policy by computing the average cost rate at different abandonment penalties. We also visualize the difference in performance using the “efficient frontier.” In particular, we normalize the class 1 abandonment penalty to unity and vary the class 2 abandonment penalty (that is, we vary $r_2/r_1$) while fixing other parameters. For each ratio, we simulate the system under our proposed policy to obtain a pair of class 1 and class 2 abandonment rates. We then obtain an efficient frontier by connecting the pair of class-based abandonment rates. For each penalty ratio, we find the best cost performance for our policy by searching over this efficient frontier (which helps mitigate the prelimit approximation error).

For our numerical study, we set the patience time distribution for class 1 to be a Beta distribution with parameters $\alpha = 4$ and $\beta = 0.1$ and for class 2 to be a Beta distribution with $\alpha = 7$ and $\beta = 0.1$; the other parameters are set at $a_1 = 0.5$, $N = 30$, $\lambda = 30$, and $\mu = 1$ (all time units in the paper are minutes unless specified otherwise). The comparison between the frontiers and the average cost rate achieved under our proposed policy and threshold control is depicted in Figure 2. The two dashed lines that wrap each bold line depict 95% confidence intervals. (For each set of penalty parameters, we repeat 50 simulations to calculate the average abandonment rates and run each simulation for one million customer arrivals. In the simulation runs, we do not allocate an early stage of simulation as a warm-up period.) For our two-class examples, class 1 static priority achieves the minimum class 1 abandonment rate and maximum class 2 abandonment rate among all feasible pairs between class 1 and class 2 abandonment rates. This corresponds to the top left corner of our frontiers. The opposite happens under class 2 static priority.

As shown in Figure 2, our proposed policy clearly outperforms threshold controls on average cost, and further, the two frontiers are clearly separated: we observe the relative improvement in average cost to vary from 1.5% to 17%. Considering the class-based abandonment rates on the efficient frontier, our proposed policy can achieve a relative improvement of more than 30%. For instance, fixing the class 1 abandonment rate at 30 per hour, our proposed policy can achieve an abandonment rate of class 2 of 27 per hour as compared with the optimal threshold control that can only achieve 40 class 2 customers abandoning per hour (with the same class 1 abandonment rate). On the other hand, when the class 2 abandonment rate is 40 per hour, our proposed policy can lower class 1 abandonment rate to 18 per hour from 30 per hour achieved under the optimal threshold control. These improvements are equivalent to a decrease in the abandonment probability of one class by an absolute 1.5% point (as opposed to a relative 1.5%) without altering the abandonment probability of the other class. This is a significant improvement in the QED regime in which the typical abandonment probability is less than 10%.

3.3. The Proposed Policy Is Robust to Moderate Changes in Parameter Misestimation

Our proposed policy is quite robust to moderate changes in system parameters. To illustrate this, we
build our proposed policy assuming a certain set of system parameters and implement it in a system that, in fact, has different parameters. That is, we solve the DCP and compute \( q^* \) using the incorrect system parameters and implement it using (4), and then we evaluate how well it performs in the actual system (that is, simulated using the correct parameters). For simulation, we use the same system parameters as in Figure 2: that is, \( \lambda = 30 \) and \( \mu = 1 \). To calculate our proposed policy, on the other hand, we assume four different settings \((\lambda, \mu) = (25, 1)\) and \((\lambda, \mu) = (35, 1)\), \((\lambda, \mu) = (35, 0.5)\) and \((\lambda, \mu) = (25, 1.5)\). The results are summarized in Figure 3(a) for \((\lambda, \mu) = (25, 1)\) and \((\lambda, \mu) = (35, 1)\) and Figure 3(b) for \((\lambda, \mu) = (35, 0.5)\) and \((\lambda, \mu) = (25, 1.5)\): the blue (in the online version) line is the performance (efficient frontier) of our proposed policy calculated under the true parameters while the red (in the online version) and the black are achieved under our proposed policy calculated under misestimated parameters. The lines in each figure are almost indistinguishable (especially compared with the performance of the threshold policy displayed in Figure 2), and so the overall average costs are also indistinguishable. In fact, we investigated our proposed policy and found that it was quite robust to such misestimation, and the difference between the corresponding \( q^* \) (calculated using misestimated information) and the true optimal \( q^* \) was very small for most of the region and had a maximum value of 18% (and the region in which the difference was close to the maximum was small). Thus, we feel that this robustness is because our proposed policy relies more heavily on the patience hazard rates, which we assume are estimated correctly in this study, rather than on the overall arrival rate and service rate parameters.

3.4. The Approach Extends to Class-Dependent Service Rates

Even for cases with class-dependent service rates, our approach can be effective. In the online appendix, we establish that solving the DCP for the class-dependent service rates case involves a complicated partial differential equation, which is not amenable to solving directly. Hence, for the case in which service rates are class-dependent, we propose to first approximate the system by setting the service rate of each class at the average weighted service across all classes, that is,

\[
\frac{1}{\mu} = \sum_{k \in K} \frac{d_k}{\mu_k},
\]

and then solve the DCP for the class-independent service rate case to compute \( q^* \) accordingly. To test the effectiveness of our approach, we compare the performance under \( q^* \) and the best threshold policy (again found by a brute force search). Figure 4 shows that our proposed policy outperforms the best threshold policy. In this example, we consider a two-class example in which \( \lambda = 30, a_1 = a_2 = 0.5, N = 30, \) and \( \mu_1 = 2, \mu_2 = 2/3, \) so that \( \mu = 1 \).

3.5. The Operating Regime Does Not Have to Be QED

Although we formally study the QED regime, our study can be used to understand the optimal control in underloaded and overloaded regimes as well. We illustrate this by comparing the performance between our proposed policy and the cost-minimizing threshold control for different server utilization levels, that is, different values of the safety staffing coefficient, \( \beta \) in (3). In particular, we consider the same two-class setting used for Figure 2 except that we vary \( \beta \in \{-4/\sqrt{30}, -2/\sqrt{30}, 0, 2/\sqrt{30}, 4/\sqrt{30}\} \), so that the number of servers, \( N \), takes values in the set \{26, 28, 30, 32, 34\}, and the corresponding traffic intensity \( \rho = \lambda / (N\mu) \) varies from 0.88 to 1.15.

For each traffic intensity, we compare the average cost obtained using our proposed policy and that obtained using the optimal threshold policy. We also
compute the frontiers for classes 1 and 2 abandonment rates using our proposed policy and using the optimal threshold policy, respectively, as before. Then, we distill these frontiers into two numbers (for each parameter) that equal the average difference between the two policies in terms of relative improvement in class 2 abandonment rate and class 2 abandonment probability, respectively, for each level of class 1 abandonment rate. Table 1 presents the results. We also report the overall average abandonment rate under the proposed policy (averaged across the classes) to give a sense of the overall quality of service at each traffic intensity.

In this numerical study, we find that the benefit of fully incorporating the entire patience-time distribution in the design of scheduling policy becomes substantial when the traffic intensity is close to 100%. For larger values of traffic intensity, our proposed policy and the cost-minimizing threshold control have similar performance as indicated by small relative and absolute improvements. On the other hand, for smaller values of traffic intensity, we observe that distinguishing our proposed policy and the cost-minimizing threshold control matters less: in this case, the fraction of abandoned customers is really small regardless of the choice of scheduling policy, which explains the huge relative improvement with the minor absolute improvement for small traffic intensities.

**Table 1.** The Proposed Policy Dominates Threshold Controls for Non-QED Regimes as Well

<table>
<thead>
<tr>
<th>Traffic intensity (ρ)</th>
<th>1.15</th>
<th>1.07</th>
<th>1.04</th>
<th>0.94</th>
<th>0.88</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average abandonment rate (proposed policy)</td>
<td>122.3</td>
<td>70.6</td>
<td>30.3</td>
<td>11.9</td>
<td>3.5</td>
</tr>
<tr>
<td>Average cost improvement (%)</td>
<td>1.9</td>
<td>8.3</td>
<td>24.3</td>
<td>56.2</td>
<td>103.3</td>
</tr>
<tr>
<td>Abandonment rate improvement (%)</td>
<td>4.2</td>
<td>21.3</td>
<td>35.2</td>
<td>62.2</td>
<td>73.2</td>
</tr>
</tbody>
</table>

4. The Diffusion Control Problem

In this section, we formulate and solve a DCP that emerges in the QED regime to approximate the optimization problem (2). In Section 4.1, we formulate that DCP by identifying the diffusion process that approximates the number of customers in the system. In Section 4.2, we use the HJB equation, which is exactly the ODE in (5), to solve the DCP and propose our scheduling policy based on the DCP solution.

4.1. The DCP Formulation

Suppose that there are no stochastic fluctuations and the system staffs under (3) with β = 0. Then, no customer will be delayed, and no server will idle because demand and supply are perfectly matched. However, with stochastic fluctuations causing inefficiency in the system, customers may be delayed and servers can idle. To formulate the DCP, we capture this inefficiency in a single-dimensional state descriptor. Let \( X(t; \pi) \) be the total number of customers in the system (either being served or waiting) centered by the number of servers at time \( t \) and under a scheduling policy \( \pi \). That is,

\[
X(t; \pi) := Q(0) + Z(0) - N + \sum_{k \in K} (A_k(t) - S_k(t; \pi) - R_k(t; \pi)),
\]

where \( Q(0) \) and \( Z(0) \) are the number of waiting customers and customers being served at \( t = 0 \), respectively; \( A_k \) and \( S_k \) are Poisson processes that count the class \( k \) arrivals and service completions, respectively; and recalling that \( R_k(t; \pi) \) counts the number of class \( k \) abandonments in \([0, t]\). More precisely, for \( Z_k(t; \pi) \) that represents the number of class \( k \) customers being served at time \( t \), as in Atar et al. (2004), \( S_k(t; \pi) = \int_0^t (\mu_k Z_k(s; \pi) \right) ds \), where \( N_k \) is a unit rate Poisson process. Observe that under any nonidling \( \pi \), \([X(t; \pi)]^+ \) and \([X(t; \pi)]^- \) count the number of waiting customers and the number of idling servers, respectively, at time \( t \).
We now find the diffusion process that approximates $X$. By respectively applying the strong approximation to $A_k$ and $S_k$ in a manner similar to that in Çelik and Maglaras (2008), $\sum_{k\in K}(A_k(t) - S_k(t; \pi))$ in (8) is decomposed into three pieces:

$$\sum_{k\in K}(A_k(t) - S_k(t; \pi)) = \left[ \sum_{k\in K}\left( a_k \lambda t - \int_0^t \mu Z_k(s; \pi) \, ds \right) \right] + \sqrt{2\lambda B(t)} + \epsilon_k(t), \quad (9)$$

where $B$ is a standard Brownian motion. On the right-hand side of (9), the first term represents the net rate at which customers are added to or subtracted from the system. By (3) and noting $N - \sum_{k\in K}Z_k(s; \pi) = [X(s; \pi)]^-$ under any nonidling policy $\pi$, this rate becomes $-\beta \sqrt{\mu \lambda t} + \mu \int_0^t [X(s; \pi)]^- \, ds$. The second term, $\sqrt{2\lambda B(t)}$, on the right-hand side of (9) captures stochastic fluctuations in customer arrivals and service completions. The last term, $\epsilon_k(t)$, on the right-hand side of (9) is an error term from the strong approximation, and it is of order of magnitude smaller than $\sqrt{\lambda t}$. By putting these three pieces together, we reach

$$X(t; \pi) = X(0; \pi) - \beta \sqrt{\mu \lambda t} + \mu \int_0^t [X(s; \pi)]^- \, ds - \sum_{k\in K} R_k(t; \pi) + \sqrt{2\lambda B(t)} + \epsilon_k(t), \quad (10)$$

for any $\pi \in \Pi$. It now only remains to express $R_k$ in (10) as a function of $X$ to derive the approximating diffusion process.

To approximate $R_k$, we need to know how much a class $k$ customer who arrived at time $t$ would wait if that customer were infinitely patient. Our best estimate of this quantity, based on a sample path version of Little’s law known as the snapshot principle (Reiman 1982), is $Q_k(t; \pi)/(a_k \lambda)$, where $Q_k$ denotes the number of class $k$ customers waiting. A critical premise for the snapshot principle to hold is that the workload configuration insignificantly changes while a customer waits (from the arrival to the service commencement). This premise holds in the QED regime because a significant change in the workload configuration resulting from customer arrivals or abandonments needs a much longer time than the amount of time a customer spends in the queue waiting for service. In a single-class version, Whitt (2005) also used $Q_k(t; \pi)/(a_k \lambda)$ to estimate how much a class $k$ customer would wait, and our approximation for $R_k$, (13), in the next paragraph, is a continuous version of equation (3.4) in Whitt (2005).

If $w$ is the amount of time a class $k$ customer would have to wait before receiving service, then $G_k(w)$ is that customer’s abandonment probability. Hence, the snapshot principle–based approximation from the previous paragraph implies that the probability a class $k$ customer who arrived at $t$ abandons the system is close to $G_k(Q_k(t; \pi)/(a_k \lambda))$. To proceed, note that for any $\pi \in \Pi$ there exists a $K$-dimensional vector of random elements, $f$, such that for any $t > 0$ we have $Q_k(t; \pi) = [X(t; \pi)]^+ f_k(t; \pi)$ where

$$(f_1(t; \pi), \ldots, f_K(t; \pi)) \in \mathcal{W}$$

Therefore, the marginal rate at which class $k$ customers abandon at time $t$ is approximately $a_k \lambda G_k([X(t; \pi)]^+ f_k(t; \pi)/(a_k \lambda))$, and so $R_k(t; \pi) \approx \int_0^t a_k \lambda G_k([X(s; \pi)]^+ f_k(t; \pi)/(a_k \lambda)) \, ds$. Observe that for any $x > 0$,

$$G_k(x) \equiv 1 - \exp\left( -\int_0^x h_k(y) \, dy \right) \approx \int_0^x h_k(y) \, dy, \quad (12)$$

where $h_k$ denotes the hazard rate function associated with the class $k$ patience-time distribution. In (12), (a) follows by a property of hazard rate function that is $G_k(x) = 1 - \exp(-\int_0^x h_k(y) \, dy)$ for $x \geq 0$ and (b) is obtained by $1 - \exp(-x) \approx x$ for small $x$ by the Taylor series expansion. Using (12), we obtain the following approximation for the cumulative class $k$ abandonment up to $t$:

$$R_k(t; \pi) \approx \lambda \int_0^t m_k\left( [X(s; \pi)]^+ f_k(t; \pi) \right) \lambda ds \quad (13)$$

recalling that $m_k$ was defined in (6). The approximation in (13) is rigorously justified in single customer class queues in heavy traffic; see Reed and Ward (2008) for a GI/GI/1+GI system and Reed and Tezcan (2012) for a GI/M/N+GI system.

Replacing $R_k$ in (10) by the right-hand side of (13), we are now ready to derive an approximating diffusion process for $X$. In particular, the approximating diffusion process, which we denote by $Z$, is given by the (weak) solution of the following stochastic differential equation:

$$Z(t) = Z(0) - \beta \sqrt{\mu \lambda t} + \mu \int_0^t [Z(s)]^- \, ds - \lambda \sum_{k\in K} \int_0^t m_k\left( [Z(s)]^+ q_k([Z(s)]^+ \right) \lambda ds + \sqrt{2\lambda W(t)}, \quad (14)$$

for some control $q \in \mathcal{W}(\infty) \equiv \{p: p(t) \in \mathcal{W} \text{ for all } t \geq 0\}$ and a standard Brownian motion $W$ independent of $B$. We assume that $Z(0) = z$ for some constant $z \in \mathbb{R}$. The control in (14) only depends on the system state—not on the time explicitly—because our focus is on stationary scheduling policies.
Using the right-hand side of (13) to replace \( R_k \) in (2) with \( X(s; \pi) \) and \( f_k(s; \pi) \) being, respectively, substituted by \( Z(s) \) and \( q_k([Z(s)])^\ast \), the objective function in (1) is approximated by

\[
V(q) = \limsup_{T \to \infty} \mathbb{E} \left[ \sum_{k \in \mathbb{X}} r_k \frac{\lambda}{T} \int_0^T m_k \left( \frac{[Z(t)]^\ast q_k([Z(t)])^\ast}{\lambda} \right) dt \right],
\]

(15)

for \( q \in \mathcal{A}(\infty) \). Then, the DCP associated with (2) is to find \( q^\ast \in \mathcal{A}(\infty) \) such that

\[
V(q^\ast) = \lim_{T \to \infty} \mathbb{E} \left[ \sum_{k \in \mathbb{X}} r_k \frac{\lambda}{T} \int_0^T m_k \left( \frac{[Z(t)]^\ast q_k([Z(t)])^\ast}{\lambda} \right) dt \right]
\]

where \( Z^\ast \) is the (weak) solution of the following stochastic differential equation:

\[
Z^\ast(t) = Z^\ast(0) - \sqrt{\mu \lambda} t + \mu \int_0^t [Z^\ast(s)]^\ast ds - \lambda \sum_{k \in \mathbb{X}} \int_0^t m_k \left( \frac{[Z^\ast(s)]^\ast q_k([Z^\ast(s)])^\ast}{\lambda} \right) ds + \sqrt{2 \lambda} W^\ast(t)
\]

for a standard Brownian motion \( W^\ast \) that is independent of the other random elements introduced in the paper (see Reed and Ward 2008, Reed and Tezcan 2012).

### 4.1.1. An Alternate DCP Formulation: The Exponential Approximation

The derivation of the DCP in (16) entails the use of the hazard rate function of the patience-time distribution. Given that delays are small in the QED regime, a more straightforward approach would be to use the value of the density function \( g_k(0) \), or \( h_k(0) \) as \( g_k(0) = h_k(0) \), to approximate the cumulative number of abandoned class \( k \) customers. This is so because

\[
m_k(x) = g_k(0)x + R_k(x),
\]

(17)

where \( R_k(x) \) is the residual term from the Taylor series expansion. Under this approach, we would define \( m_k(x) \) as \( m_k(x) := g_k(0)x \) instead of the one defined following (5), and (13) would change to

\[
R_k(t; \pi) \approx \int_0^t g_k(0)[X(s; \pi)]^\ast f_k(s; \pi) ds,
\]

(18)

and the resulting diffusion process would have linear drift. The approximation in (18) for single customer class queues in heavy traffic is proven to be accurate in the literature. Note that (13) and (18) are identical if and only if the class \( k \) patience-time distribution is exponential. Henceforth, we call the approximation in (18) "the exponential approximation."

### 4.2. The DCP Solution

Our solution approach for the DCP in (16) is to use stochastic calculus by applying Itô's lemma on the stochastic process \( Z \) in (14); see Øksendal (2003). This leads to the optimality equation given in (5) that is commonly referred to as the HJB equation.

We call a pair \((v, \kappa)\) of a continuously differentiable function, \( v: (-\infty, d) \to \mathbb{R} \), and a positive constant, \( \kappa \), that solves (5) a solution of the HJB equation. Using the HJB equation, we derive an optimal control of (16) in the following theorem. The theorem also provides some properties of the HJB solution that are useful in characterizing the structure of an optimal control of the DCP in (16).

**Theorem 1.** (i) There exists a continuously differentiable function \( v^\ast \) and a positive constant \( \kappa^\ast \) such that the pair \((v^\ast, \kappa^\ast)\) is a unique solution of the HJB equation in (5). Furthermore, \( \sup_{x,d} |v^\ast(x)| \leq \min\{r_1, \ldots, r_k\} \) and \( v^\ast \) is increasing.

(ii) For the DCP in (16), \( \kappa^\ast = \inf_{q \in \mathcal{A}(\infty)} V(q) \) and \( q^\ast(x) := \arg \min_{q \in \mathcal{A}(\infty)} \phi(x, v^\ast(x), q) \) for \( x < d \) is an optimizer.

Theorem 1 connects the structure of an optimal control of the DCP and the underlying patience-time distributions through (7), which we reproduce here for the reader’s convenience:

\[
q^\ast(x) = \arg \min_{q \in \mathcal{A}} \left\{ \sum_{k \in \mathbb{X}} r_k m_k \left( \frac{x^+ q_k}{\lambda} \right) - \sum_{k \in \mathbb{X}} v^\ast(x) m_k \left( \frac{x^+ q_k}{\lambda} \right) \right\}.
\]

We see from this relation that the DCP solution balances the short-term and the long-term consequences in making scheduling decisions. The first term on the right-hand side of this expression arises from the integrand of the objective function in (16), and it represents the instantaneous (or marginal rate of) abandonment cost. The second term, on the other hand, reflects future consequences of scheduling decisions made at the present through the value function, \( v^\ast \). Therefore, \( q^\ast \) is designed to take both the short-term and long-term consequences into consideration. This is an important aspect of optimal scheduling when catering to impatient customers.

### 5. Static Priority and Threshold Control

In this section, we derive a solution of the DCP (16) when \( m_k(x) \) is linear in \( x \) for all \( k \in \mathbb{X} \), which happens under the exponential approximation in (18). The optimal control from this linear case will be a benchmark for optimal controls derived under the more general approximation (13) (both performance-wise and structure-wise). In the linear case, the HJB solution takes the form of threshold control, which is defined as follows.
**Definition 1.** Let $\mathcal{D} \subseteq \mathbb{R}$, with $|\mathcal{D}| = J \geq 2$, and let $(p_1, \ldots, p_J)$ be a permutation of the class indices in $\mathcal{D}$. A threshold control is defined by a $J - 1$ dimensional vector $L = (L_1, \ldots, L_{J-1})$ having $L_0 := 0 < L_1 < \cdots < L_{J-1} < L_J := \infty$ such that

$$q_{p_j}(x) = 1_{\{L_{j-1} \leq x < L_j\}}, \quad \text{for } j \in \mathcal{D}$$

and

$$q_{p_j}(x) = 0, \quad \text{for } j \notin \mathcal{D}.$$ 

From Definition 1, we see that the key in the design of a threshold control is to specify the class that has the lowest priority as a function of the system workload. The full priority ranking among the classes is left unspecified. This is because, in heavy traffic, we expect that the class that has the lowest priority will be the only class having nontrivial delay, and so the full priority ranking among the classes other than the lowest priority would be redundant. The static priority version of a threshold control is defined by setting $J = 1$ in Definition 1. Then, the set $\mathcal{D}$ contains a single element $k$ having $q_k(x) = 1$ for $x > 0$.

We next describe an algorithm that constructs an optimal threshold control for a general $K$-class setting in which, for each $k \in K$, we have $m_k(x) = \gamma_k x$ for some constant $\gamma_k > 0$ and all $x > 0$. In a $K$-class setting, an optimal threshold control may have a set of classes that never have the lowest priority. Our algorithm identifies this set of classes and also computes the threshold levels at which the remaining classes will have the lowest priority. Our algorithm consists of four steps, and we illustrate each step through a six-class example in Figure 5. After describing the algorithm, we state the formal result that establishes the optimality of the so-constructed threshold control in Proposition 1. For convenience, we will assume the following, which is without loss of generality:

$$\gamma_1 r_1 \geq \gamma_2 r_2 \geq \cdots \geq \gamma_K r_K.$$

**Figure 5.** Visualization of the Algorithm to Compute the Optimal Threshold Control for a Six-Class Example
5.1. Algorithm to Find an Optimal Threshold Control

1. Define

\[ f_j := \{ k \in \mathbb{R} : \gamma_k > \gamma_j \} \text{ for all } j \in \mathbb{R} \text{ such that } j > k \].

Then, for any \( k \in \mathbb{R} \setminus f_j \), we set \( q_j^*(x) = 0 \) for \( x > 0 \).

Observe that \( f_j \) is nonempty because \( K \in f_j \) and, further, that the classes in \( f_j \) are such that, for any \( i, j \in f_j \) with \( i < j \), we have \( \gamma_i > \gamma_j \) (by definition of this set) and \( \gamma_i r_i \geq \gamma_j r_j \) (by the initial assumption).

Intuition. For \( k, j \in \mathbb{R} \), suppose \( k > j \) and \( \gamma_k \geq \gamma_j \). Then, class \( k \) has both a lower instantaneous abandonment cost (because \( \gamma_k r_k \leq \gamma_j r_j \)) and a higher abandonment rate. That is, prioritizing class \( j \) over class \( k \) reduces the system congestion at a faster rate with lower penalties compared to the other way around, implying \( q_j^*(x) = 0 \) for \( x > 0 \).

Example. For our six-class example depicted in Figure 5, applying step 1, we obtain \( f_1 = \{1, 2, 4, 5, 6\} \). That is, we remove class 3 in step 1 (as shown in Figures 5(a) and (b)) because \( \gamma_3 < \gamma_k \) for \( k \in \{4, 5, 6\} \).

2. Define \( i_1, i_6 \) as the lowest and highest numbered classes in \( f_j \), respectively; that is, \( i_1 = \min\{j : j \in f_j\} \) and \( i_6 = \max\{j : j \in f_j\} \) for each \( j \). We construct the line between classes \( i_1 \) and \( i_6 \) and identify all classes lying to the right of that line by defining

\[ \mathcal{C} := \{ (\gamma_j, \gamma_j r_j) : j \in f_j \} \cap \left\{ (x, y) \in \mathbb{R} \times \mathbb{R} : y \leq \frac{\gamma_i r_i - \gamma_j r_j}{\gamma_i - \gamma_j} (x - \gamma_i) + \gamma_j r_j \right\} \]

and \( f_2 := \{ j \in f_1 : (\gamma_j, \gamma_j r_j) \in \mathcal{C} \} \). (Observe that \( f_2 \) is nonempty because \( i_1, i_6 \in f_2 \)). Then, for any \( j \in f_1 \setminus f_2 \), we set \( q_j^*(x) = 0 \) for \( x > 0 \).

Intuition. For \( k < j \), \( (\gamma_k r_k - \gamma_j r_j)/(\gamma_k - \gamma_j) \) captures the marginal benefit of prioritizing class \( k \) customers over class \( j \) customers. By serving a class \( k \) customer over a class \( j \) customer, the system can reduce the instantaneous abandonment cost, which is myopic, by \( \gamma_k r_k - \gamma_j r_j \). On the other hand, by prioritizing class \( j \) over class \( k \), the system gains the additional abandonment rate of \( \gamma_k - \gamma_j \) and hence can better trim down the congestion, which is forward-looking. This step can be considered to be an extension of step 1 because, in this step, we are eliminating classes by comparing them with a convex combination of classes \( i_1 \) and \( i_6 \) rather than comparing them with individual classes as in step 1.

Example. We remove class 2 in step 2 (as shown in Figures 5(b) and (c)) because class 2 lies to the left of the line defining \( \mathcal{C} \). Then, we obtain \( f_2 = \{1, 4, 5, 6\} \).

3. Let \( f_3 := \{ j \in f_2 : (\gamma_j, \gamma_j r_j) \notin \text{int}(\text{conv}(\mathcal{C})) \} \), where, for any set \( A, \text{int}(A) \) and \( \text{conv}(A) \) denote the interior and the convex hull of the set \( A \), respectively. For any \( j \in f_2 \setminus f_3 \), we set \( q_j^*(x) = 0 \) for \( x > 0 \).

Intuition. For any \( j \in f_2 \setminus f_3 \), we can always find \( j, j \in f_2 \) such that the \( (\gamma_j, \gamma_j r_j) \) is above the line that connects \( (\gamma_k r_k, \gamma_k r_k) \) and \( (\gamma_k r_k, \gamma_k r_k) \). Hence, we can repeat the same argument from step 2 and conclude that \( q_j^*(x) = 0 \) for \( x > 0 \). Intuitively, this extends the logic of the previous step by comparing classes with convex combinations of multiple classes, instead of two classes as in step 2.

Example. We remove class 4 (as shown in Figures 5(c) and (d)) because class 4 lies in the convex hull of \( \mathcal{C} \). Thus, we obtain \( f_3 = \{1, 5, 6\} \).

4. Optimize the threshold structure within \( f_3 \). We use the notation \( [j] \) to refer to the class with the \( j \)th smallest index in the set \( f_j \); for instance, \( [1] = \min\{i : i \in f_j\} \), and \( [j] = \max\{i : i \in f_j\} \), where \( J_i := |f_i| \). Using this notation, we define \( T_{j, j+1} \) as

\[ T_1 := \infty, \quad T_j := \frac{r_{[j-1]} - r_{[j]} - r_{[j]} x}{y_{[j]} - y_{[j]}}, \quad \text{for } j \in \{2, \ldots, 3\}, \]

\[ T_{j, j+1} := 0. \]

Then, for \( x > 0 \), we set

\[ q_{[j]}^*(x) = 1_{[T_{j, j+1} < x]} x. \]

It is straightforward to verify that \( q \) given in this reduction algorithm is a threshold control because \( T_j \) is decreasing in \( j \) (because \( f_j \) is the set of vertices of a convex set) and \( y_{i}^*(x) \) is increasing in \( x \) by Theorem 1.

Example. The optimal control is given by

\[ q_{[1]}^*(x)^* = 1_{[T_{1, 2} < x]} x, \quad q_{[2]}^*(x)^* = 1_{[T_{1, 2} < x]} x, \quad q_{[3]}^*(x)^* = 0, \quad \text{for } j \in \{1, 5, 6\}, \]

for \( x > 0 \) and \( j = 1, 2, 3 \), which uses our notation \( [1] = 1 \), \( [2] = 5 \), and \( [3] = 6 \) for the classes in \( f_3 \).

We next formalize the optimality of the threshold control that we have constructed in the following proposition.

Proposition 1. Suppose \( m_j(x) \) in (15) equals \( \gamma_j x \) for some \( \gamma_j > 0 \) and all \( k \in \mathbb{R} \) with \( \gamma_i r_i \geq \gamma_j r_j \geq \gamma_k r_k \). Then, either a static priority or a threshold control as defined by \( q^* \) in (19) is optimal for the DCP (16).

Note that by (19), if \( f_3 = \{k\} \) for some \( k \in \mathbb{R} \), then the optimal control is a static priority and given by \( q_j^*(x) = 1 \) for \( x > 0 \).

When the patience-time distributions are all exponential, Proposition 1, in fact, suggests the asymptotic optimality of threshold control or static priority. To see this, note that the HJB in this exponential case was studied in Section 5 of Atar et al. (2009) (with a caveat that they considered the infinite horizon discounted cost criterion whereas we look at the long-run average cost criterion). The paper by Atar et al. (2009) proves...
the asymptotic optimality of the HJB solution while not explicitly solving the HJB equation. Hence, the asymptotic optimality of threshold control or static priority would follow by extending the result in Atar et al. (2009) to the infinite horizon case and then combining it with Proposition 1 of this paper.

Even for nonexponential patience-time distributions, Proposition 1, together with the accuracy of the exponential approximation found in the previously mentioned single-class papers, suggests that threshold control or static priority should perform well. However, there are two major concerns. First, there are many distributions having \( g_k(0) = 0 \) or \( g_k(0) = \infty \), in which case the exponential approximation is no help. Second, in comparison with the mean of a distribution, the value of the density function (or its derivative of any order in general) at zero is not a robust statistic, which makes relying on that one point estimation somewhat worrying. Therefore, from the policy design perspective, it is preferable to understand the solution of (16) when the entire patience-time distribution is included and ask whether or not the solution is a threshold control or static priority.

Unfortunately, threshold control or static priority are not optimal in general for the DCP (16). We begin with an example in which the solution to the HJB equation is neither a threshold nor a static priority control.

Proposition 2. Suppose \( g_k(x) \) is continuous for \( k \in \mathcal{K} \), and defining,

\[
\mathcal{K}_n := \{ k \in \mathcal{K} : g_k(0) = 0 \text{ and } g_k(x) \text{ is strictly increasing at } x = 0 \},
\]

we have \( |\mathcal{K}_n| \geq 2 \) and \( g_k(0) > 0 \) for \( k \in \mathcal{K} \setminus \mathcal{K}_n \). Then, \( q^* \) is neither a threshold nor a static priority control. Furthermore, if \( \beta \geq 0 \), neither threshold control nor static priority is optimal for the DCP (16).

The stated conditions in the proposition exclude situations in which \( g_k(x) \) is decreasing in \( x \) near the origin for all \( k \in \mathcal{K} \). In such situations, it turns out that we can identify conditions under which threshold control or static priority are, in fact, optimal (see Section 6.1). So, although static priority and threshold control are not optimal in general, there are still cases in which they are. Note that we can relax the condition \( \beta \geq 0 \) under mild assumptions on the hazard rate function; see the last paragraph of the proof.

Thus far, we have established that threshold control and static priority are not optimal in general. We would like to next better understand the conditions under which this suboptimality occurs and, further in those cases, better understand the structure of the optimal policy. We do so in the next section by focusing on a two-class system, which is more analytically tractable.

5.2. Threshold Control or Static Priority Does Not Solve HJB Equation

For \( k \in \mathcal{K} \), let \( r_k = r > 0 \), and let the hazard rate of the class \( k \) patience-time distribution be continuous and strictly increasing without bound in its domain. Then, \( q^* \) is given by

\[
q^*(x) = \left( \frac{\lambda_a}{x} h_1^{-1} \left( \frac{\psi(x)\lambda}{x} \right), \ldots, \frac{\lambda_a}{x} h_K^{-1} \left( \frac{\psi(x)\lambda}{x} \right) \right), \quad (20)
\]

for all \( x \in (0, d) \) where, for each \( x \in (0, d) \), \( \psi(x) \) is a unique constant that satisfies

\[
\sum_{k \in \mathcal{K}} \frac{\lambda_a}{x} h_k^{-1} \left( \frac{\psi(x)\lambda}{x} \right) = 1.
\]

Because the hazard rate function for each class is assumed to be continuous, \( q^* \) in (20) is continuous in \( x \) and, hence, is neither a threshold control nor a static priority.

However, solving the HJB Equation (5) is only a sufficient condition for the DCP solution (16) and not necessary. The following proposition establishes that both threshold control and static priority are suboptimal in the previous example—and in any other example that satisfies the conditions of the proposition.

6. A Two-Class Model

In a two-class setting, the optimal control in Theorem 1 reduces to

\[
q_1^*(x) = \arg \min_{q_1 \in [0, 1]} \phi(x, v^*(x), (q_1, 1 - q_1)), \quad \text{for all } x > 0, \quad (22)
\]

and \( q_2^* = 1 - q_1^* \). The case \( x \leq 0 \) corresponds to the system having idle servers, meaning the scheduling policy becomes irrelevant. In particular, when \( x \leq 0 \), \( \phi(x, w, (q_1, 1 - q_1)) = 0 \) for any \( w \in \mathbb{R} \) and \( q_1 \in [0, 1] \), so in this section, we only consider \( x > 0 \).

Our objective in this section is to understand how the patience-time distribution influences the structure of \( q^* \). In particular, we analyze the solution of (22) for different types of the class 1 patience-time distribution while fixing the class 2 patience-time distribution to be exponential with rate \( \gamma_2 \). First, in Section 6.1, we provide conditions under which a decreasing hazard rate function leads to the optimality of the static priority and threshold structures. Then, in Section 6.2, we find that without the decreasing hazard rate function, the optimal control has a U-shape structure (and so is neither threshold nor static priority).
To state the results in the rest of the paper, we denote threshold controls by
\[
q^1_T(x) :=
\begin{cases}
(0, 1) & \text{if } x < T, \\
(1, 0) & \text{if } x \geq T,
\end{cases}
\text{ for } T \in (0, \infty),
\]
\[
q^2_T(x) :=
\begin{cases}
(1, 0) & \text{if } x < T, \\
(0, 1) & \text{if } x \geq T,
\end{cases}
\text{ for } T \in (0, \infty),
\]
and we denote static controls by
\[
q^1_S(x) := (0, 1) \quad \text{for all } x > 0,
\]
\[
q^2_S(x) := (1, 0) \quad \text{for all } x > 0,
\]
so that under \(q^k_S\), class \(k\) customers are statically prioritized over the other class customers.

6.1. Optimality of the Threshold and Static Priority
Suppose the hazard rate function associated with the class 1 patience-time distribution is decreasing. (Note that this implies the patience-time distribution must have unbounded domain.) If \(\gamma_2\) is small compared with \(h_1(0)\), then, to minimize abandonments, intuition suggests prioritizing class 1 customers when the number of customers in the system is small. This is because the class 2 customers are likely to be patient enough for the few class 1 customers in the system to complete their service. However, as the number of customers increases, depending on the penalties \(r_1\) and \(r_2\), we may want to take advantage of the fact that any class 1 customer who has spent some time waiting becomes even more likely to wait longer since \(h_1\) is decreasing. This suggests that the priority could switch, meaning the structure of the optimal policy will be threshold. If the priority never switches, then the structure is static priority with class 1 customers always having priority over class 2 customers.

We formalize the intuition for the threshold and static priority structures in the preceding paragraph by providing necessary and sufficient conditions under which these structures emerge. To do this, we first observe that when \(h_1\) is decreasing, then \(\phi(x, \nu^*(x), (q_1, (1 - q_1)))\) is concave in \(q_1 \in [0, 1]\), which means that the minimum occurs at a corner point. Because the class 1 patient-time distribution has unbounded support, the corner points for \(q_1\) are 0 and 1. Thus, it is straightforward to verify that we have
\[
q^*(x) =
\begin{cases}
(0, 1) & \text{if } r_1 - \nu^*(x) \frac{m_1(x/\lambda)}{r_2 - \nu^*(x)} \geq \gamma_2, \\
(1, 0) & \text{otherwise},
\end{cases}
\]
(23)
The conditions for the threshold and static priority structures emerge from (23) when \(r_2 \geq r_1\) by recognizing that \((r_1 - \nu^*(x))/(r_2 - \nu^*(x))(m_1(x/\lambda)/(x/\lambda))\) is decreasing in \(x\). Proposition 3 formally establishes this result.

**Proposition 3.** Suppose \(h_1(x) \leq h_1(y)\) for any \(x > y \geq 0\) and \(h_2(x) = \gamma_2\) for \(x \geq 0\).
(i) Suppose \(r_2 > r_1\). Then, there exists \(T \in (0, \infty)\) such that \(q^* = q^1_T\) if and only if
\[
(r_1 - \nu^*(0))h_1(0) > (r_2 - \nu^*(0))\gamma_2.
\]
(ii) Suppose \(r_2 = r_1\) and \(h_1(0) > \gamma_2\). Then, there exists \(T \in (0, \infty)\) such that \(q^* = q^2_T\) if and only if
\[
\lim_{x \to \infty} \frac{m_1(x)}{x} < \gamma_2.
\]
(24)
(25)

Otherwise, if (24) does not hold, then \(q^* = q^1_T\).

The optimality of threshold control does not generalize. Even when the hazard rate function associated with the class 1 patience-time distribution is decreasing, if \(r_1 > r_2\), it is, in general, not true that \(q^*\) takes threshold structure as in the case in which \(r_2 \geq r_1\). If \(r_1 > r_2\), prioritizing class 2 customers by relying on class 1 customers’ increasing patience might not be optimal when the number of waiting customers is large. Especially if class 1 customers are not sufficiently patient as their waits increase, we can formalize that it is optimal to prioritize class 1 customers who have waited enough. The consequence is the suboptimality of threshold control even under the same condition, \((r_1 - \nu^*(0))h_1(0) > (r_2 - \nu^*(0))\gamma_2\) from Proposition 3(i). This suboptimality is formalized in the following result.

**Proposition 4.** Suppose \(r_1 > r_2\) and \(h_1\) is a nonconstant function with \(h_1(x) \leq h_1(y)\) for any \(x > y \geq 0\) and \(h_2(x) = \gamma_2\) for \(x \geq 0\). If \((r_1 - \nu^*(0))h_1(0) > (r_2 - \nu^*(0))\gamma_2\), and \(\lim_{x \to \infty} m_1(x)/x > 0\), then there exist constants \(0 < T_1 \leq T_2 < \infty\) such that \(q^*(x) = (0, 1)\) for \(x \in (0, \infty)\setminus[T_1, T_2]\), and \(q^*(x) = (1, 0)\) otherwise. Furthermore, if \(\beta \geq 0\), \(V(q^*) > V^\star\) for \(i \in \{1, 2\}\) and any \(T \in (0, \infty)\).

In the statement of Proposition 4, the condition \(\lim_{x \to \infty} m_1(x)/x > 0\) corresponds to the situation in which class 1 customers are not sufficiently patient as they wait. Note that in Proposition 4, the case \(T_1 = T_2\) (in which case the control is static priority) is neither completely ruled out nor are we able to provide sufficient conditions under which this may happen. This is because, when \(r_1 > r_2\), \(\lim_{x \to \infty} m_1(x)/x \geq \gamma_2\) implies \(q^*(x) = (0, 1)\) for all \(x > 0\) by (23). Also, sufficient conditions to exclude \(T_1 = T_2\) heavily depend on the structural properties of \(\nu^*\) for which only limited knowledge (the ones given in part (i) of Theorem 1) is available.

6.2. Structure of Optimal Policy When Threshold Control and Static Policy Are Suboptimal
When the hazard rate associated with the class 1 patience-time distribution is not decreasing, the structure of the optimal control is less straightforward.
to describe analytically. Therefore, we begin with a numerical study. Figures 6(a) and 6(b), respectively, solve for the optimal control when the class 1 hazard rate is Weibull \((r_1 = 3, r_2 = 3.3, h_1(x) = 0.4x^3, \lambda = 100, \gamma_2 = 0.5, \beta = 0, \mu = 1, \text{ and } \alpha_1 = 0.2)\) and also when it is log-normal with mean 1 and standard deviation 0.5 \((r_1 = 3, r_2 = 3.3, \lambda = 100, \gamma_2 = 0.1, \beta = 0, \mu = 1, \text{ and } \alpha_1 = 0.6)\). In both cases, the optimal control has a U-shape structure.

The intuition for the shape of \(q^*_1\) in Figure 6 is as follows. When the number of customers waiting is small (workload \(x\) close to 0), since the class 1 hazard rate is small around 0 (true for both the Weibull and log-normal distributions that we consider for which \(h_1(0) = 0\)), we can safely prioritize class 2 customers and rely on class 1 customers being patient enough to wait for some class 2 customers to finish their service. However, since the class 1 hazard rate is increasing near the origin as the number of customers waiting increases, class 1 customers become more impatient and begin abandoning in greater numbers. Then, even though class 1 has the lower penalty per abandonment than class 2 does, \(r_1 < r_2\), as in the setting for Figure 6, the cost incurred by class 1 abandonment will be significant. The fix is to not strictly prioritize class 2 customers and to split priorities between the two classes as the number of waiting customers increases. This explains why \(q^*_1\) decreases in Figure 6. However, eventually we prefer to let class 1 customers abandon because either (i) (increasing hazard rate) we want to take advantage of their high abandonment rate to lower overall system congestion at a lower penalty per abandonment \((r_1 < r_2)\), or (ii) (unimodal hazard rate) we want to take advantage of their decreased probability to abandon after waiting large amounts of time, consistent with Proposition 3. If the hazard rate is always increasing (as in the case for the Weibull in Figure 6(a)), we can do this in a manner that continuously balances the instantaneous cost of abandonments and the future benefit. If the hazard rate is decreasing for large workload \(x\) (as is the case for the log-normal, which is unimodal, in Figure 6(b)), continuously achieving the aforementioned balance is not possible, and \(q^*_1\) becomes discontinuous as in Propositions 3 and 4.

Figure 6 and the intuition behind it make it clear that we do not expect the optimal policy to have a threshold structure when the class 1 hazard rate is neither decreasing nor constant. Proposition 5 formalizes this discussion by characterizing the optimal control for the case in which class 1 hazard rate is increasing.

**Proposition 5.** Suppose \(h_1\) is a nonconstant function, and \(h_1(x) \geq h_1(y)\) for any \(x > y \geq 0\) and \(h_2(x) = \gamma_2\) for \(x \geq 0\). Then, \(q^*\) is given by

\[
q^*(x) = \min\left\{1, \frac{\lambda_1}{x} h_1^{-1}\left(\frac{r_2-v^*(x)}{r_1-v^*(x)}\right)\right\},
\]

\[
1 - \min\left\{1, \frac{\lambda_1}{x} h_1^{-1}\left(\frac{r_2-v^*(x)}{r_1-v^*(x)}\right)\right\}
\]

for \(x \in (0, d)\), and \(V(q^*_1) > \kappa^*\) and \(V(q^*_2) > \kappa^*\) for \(i \in \{1, 2\}\) and \(T \in [0, \infty)\). Furthermore, \(V(q^*_3) > \kappa^*\) if

(i) \(r_1 \geq r_2\) or
(ii) \(r_1 < r_2\) and \(G_\epsilon\) has the bounded domain.

7. Conclusion

In this paper, we study the problem of scheduling heterogeneous customers to minimize long-run average abandonment costs in a many-server service system. Such systems have been studied for cases in which customers have exponential patience distributions, and our focus is on the general patience distribution case. We use a diffusion-based approach to characterize the
Hamilton–Jacobi–Bellman optimality equations that a near-optimal control should satisfy, and further, we propose a near-optimal policy based on the solution to these HJB equations. Even for exponential patience distributions, the structure of the solution to the HJB equations has not been identified in the literature, and our first contribution is to provide this characterization: we formally prove that a threshold control policy is asymptotically optimal. For nonexponential patience-time distributions, by analyzing the HJB equations, we find that the optimality of threshold control does not generalize. We use a two-class setting to better understand the structure of the near-optimal scheduling policy. We find that under some technical conditions, this policy has a novel U-shaped structure that prioritizes one class for low and high workloads and the other class for intermediate workloads. Further, our numerical studies show that compared with threshold control, our proposed policy is able to reduce abandonment rates of one class by 30% without affecting the abandonment rate of the other class and lower average abandonment costs by up to 17%.

We also investigated the performance of fixed-queue ratio (FQR) class of controls (see Gurvich and Whitt 2010), which tend to be easy to compute and implement. The FQR controls have state-independent $q^*$ values, that is, $q^*(x) = (q_1^*, \ldots, q_K^*) \in \mathbb{R}_+^K$ such that $q_k^* \geq 0$ and $\sum_{k=1}^{K} q_k^* = 1$. FQR can be implemented by substituting $q^*$ in (4), which, in general, could be state-dependent, by the state-independent ratios $q^*$. Using numerical studies, we observed that, if $g_k^*(0) = 0$ for all $k \in \mathcal{K}$, then an FQR control outperforms the cost-minimizing threshold control and has performance very close to that of our proposed policy. However, when $g_k^*(0) > 0$ for some $k \in \mathcal{K}$, then the FQR controls perform similarly to threshold controls and are dominated by our proposed policy. Further, we observed that even if an FQR control outperforms threshold control, the performance gap is small. Thus, noting that computing optimal threshold controls can be very involved, especially if the number of classes is large, our numerical studies support the class of state-independent FQR controls as a more efficient scheduling heuristic than the class of threshold controls. We also believe that understanding the performance of state-independent FQR controls in more generality is an important topic for additional research.

We feel there are several additional avenues for future work in this area. In the paper, we focus on a two-class setting to generate greater insight into the structure of near-optimal scheduling policies for general patience distributions. A natural direction to extend this work would be to consider the general $K$-class setting and characterize the near optimal scheduling policies directly. Our numerical results indicate that the FQR control dominates threshold control and may have performance that is close to the near-optimal policy for two classes. Formalizing these observations analytically and extending them to $K$ classes would also make for an interesting study. Finally, other avenues worth pursuing would be to study optimal scheduling (i) when customer patience distributions are not a priori known but are estimated in an online manner while scheduling simultaneously, (ii) when customers are strategic and change their abandonment behavior based on the scheduling policy, and (iii) when the customer abandonment behavior can be changed by the system by potentially providing messages to customers about their anticipated wait times.

References


