# The Power of Letting Go: Scheduling with Immediate and Delayed Rejections in Multi-Server Queues

Arsenii Shmelev, Noa Zychlinski
Faculty of Data and Decision Sciences,
Technion – Israel Institute of Technology, Haifa 3200003, Israel
ORCID: 0000-0002-5125-3089, noazy@technion.ac.il

Long waiting times in service systems reduce customer satisfaction, increase abandonment rates, and harm provider performance. To address these challenges, we study a multi-server, multi-class queueing system in which the operator can jointly control scheduling and implement two types of rejections: *immediate rejections* at arrival and *delayed rejections* during the waiting period. The objective is to minimize total operational costs, including waiting costs and rejection penalties.

Using a fluid approximation, we characterize the structure of the optimal policy and develop an index-based control rule—the $\mathcal{L}\mu$ rule—that jointly governs scheduling and rejection decisions. Under the $\mathcal{L}\mu$ rule, each customer class operates in one of two distinct queueing regimes: an Erlang-B regime, where customers are rejected immediately if capacity is unavailable, or an Erlang-A regime, where customers are admitted and may be rejected after waiting.

We further extend the model to incorporate class-specific service-level (SL) requirements. In this setting, we show that the index structure adjusts to ensure SL compliance, relaxing strict scheduling priorities and enabling partial capacity sharing across classes. Delayed abandonments, which play a limited role in the basic model, become effective under the adjusted policy due to the presence of SL constraints. Numerical simulation experiments illustrate the effectiveness of the proposed policies.

*Key words*: scheduling queues, queues with rejection, service-level requirements, fluid models

## 1. Introduction

Long waiting times, a frequent issue in many service and healthcare systems, diminish customer satisfaction, increase the likelihood of customer abandonment, and can negatively impact both provider profitability and reputation (Jones and Peppiatt 1996, Silvester et al. 2004, Michael et al. 2013, Bleustein et al. 2014, Worthington 2004). To mitigate these effects, service providers can increase their service capacity or staffing levels (Harrison and Zeevi 2005, Whitt 2006b), though this is not always feasible or cost-effective. Another option is to improve scheduling processes to prioritize certain customer classes over others (Atar et al. 2010, Huang et al. 2015).

Alongside scheduling, we optimize two types of rejections: *immediate rejections*, which occur at the time of arrival, and *delayed rejections*, which occur after a customer has waited in the system

(i.e., controlled waiting-time–based removals of customers who have not yet entered service). Many real-world systems allow customers to wait provisionally, only to be removed later if they are not served within a reasonable time, as seen in practices such as waitlist expirations, callback timeouts, or virtual holding queues.

In other words, immediate rejections control who is allowed to enter the system, while delayed rejections control how long a customer is permitted to stay. Together, they enable policy flexibility while maintaining analytical tractability. In particular, the delayed rejection mechanism introduces time-dependent control, which is essential for meeting service-level requirements and dynamically managing congestion. The rationale behind these two strategies is to regulate customer flow more effectively and maintain a high service level for those currently being served.

Specifically, these mechanisms allow the system operator to redirect certain customer classes either upon arrival or while they are waiting by dispatching them to a different service provider. In healthcare, for example, patients might be sent to an alternative healthcare unit/provider. In call centers arriving customers or customer who have been waiting for an extended period may be removed from the system, effectively ending their waiting time without receiving the intended service. By redirecting or removing customers, the system can reduce congestion, optimize resource utilization, and improve overall service efficiency. These strategies also highlight the trade-offs between service availability and system efficiency, as some customer classes will receive a better service level, while others may receive worse service or none at all from the specific provider.

Several applications are relevant to our model, where implementing immediate rejections (at arrival) or delayed rejections (after admission but before service) is both reasonable and beneficial. In healthcare management, particularly in emergency departments or specialized clinics, immediate rejection or redirection of non-critical patients ensures that critical cases receive priority care, while delayed rejection may occur when patients are initially admitted but later diverted after waiting too long. In customer service call centers, lower-priority calls may be blocked immediately, or callers may be placed on hold and disconnected later if wait times exceed acceptable limits. In manufacturing, production systems may reject orders upon submission to avoid overload or cancel pending orders that have waited too long. In IT and cloud services, non-essential users may be denied access up front or disconnected dynamically to preserve performance for critical users. In the airline industry, immediate rejection occurs at the time of booking when flights are full, but more commonly, delayed rejections are used: passengers with valid tickets are admitted and wait, but some are later asked to give up their seat—often with compensation—when overbooking is resolved at the gate. Lastly, in the hospitality industry, hotels may decline last-minute reservations or overbook intentionally and reassign guests at check-in, balancing immediate and delayed rejection strategies to accommodate high-value or loyal customers.

The paper's contribution can be summarized as follows:

- *Modeling and Problem Formulation.* We study a multi-class, multi-server queueing system, motivated by applications in healthcare, customer service, and cloud operations, where each class must meet an SL requirement. To address this problem, we formulate a joint scheduling and admission control problem that incorporates two distinct rejection mechanisms: *immediate rejections*, applied probabilistically at arrival, and *delayed rejections*, applied to customers waiting too long in queue. This dual mechanism extends classical admission control models by enabling greater flexibility in meeting service and cost objectives.
- *Main Structural Result: The $\mathcal{L}\mu$ Rule.* We show that the optimal policy for the baseline model takes the form of an index-based $\mathcal{L}\mu$ rule, which balances each class's holding cost, service rate, abandonment rate, and both rejection costs. Under this policy, at most one class is partially rejected (with probability $p_i \in (0,1)$), while all others are either fully admitted or fully rejected. Conceptually, the rule partitions classes into two queueing regimes: Erlang-B (immediate rejections) and Erlang-A (delayed rejections), depending on the optimal rejection behavior.
- *Impact of SL Constraints: The $\mathcal{L}^S\mu$ Rule.* We show that incorporating SL constraints alters the optimal policy structure. The $\mathcal{L}\mu$ rule is adapted into the $\mathcal{L}^{\mathrm{S}}\mu$ rule, which relaxes strict priority and reallocates capacity across classes to satisfy SL constraints. In particular, classes that would otherwise be rejected under the $\mathcal{L}\mu$ rule may receive capacity at the expense of higher-index classes, ensuring compliance with SL thresholds. Importantly, we show that the delayed rejection mechanism, which plays a limited role in the basic model, becomes operationally meaningful in the presence of SL constraints, allowing the system to control waiting times in a class-specific and cost-effective way. Translating the fluid solution back into the original stochastic setting yields a structured, two-stage policy that balances cost-efficiency with service-level feasibility.

The rest of the paper is organized as follows. Section 2 provides a brief review of the relevant literature. In Section 3, we introduce the stochastic model and the optimization problem. Section 4 presents the fluid model analysis, the optimality of the $\mathcal{L}\mu$ rule, and the translation of the policy back to the stochastic system. In Section 5, we integrate class-specific SL requirements and introduce the $\mathcal{L}^{\mathrm{S}}\mu$ rule and its translation back to the stochastic system. Finally, Section 6 concludes the paper and suggests several directions for future research. All proofs are provided in the appendix.

## 2. Literature Review

This paper is related to three primary areas of literature: the scheduling and resource allocation of queues with multiple customer classes, queueing systems with rejection, and service level with service level (SL) requirements. We will provide a brief review of the relevant literature in these areas.

*Scheduling and resource allocation of queues with multiple customer classes.* The scheduling of multiple customer classes in stochastic processing networks has been extensively studied. A foundational result in this literature is the $c\mu$ rule, introduced by Cox and Smith (1961), which establishes the optimality of a simple index-based policy for single-server queues with linear holding costs. Subsequent research has generalized this result to various settings, although these extensions are typically shown to be optimal only in asymptotic regimes (e.g., Van Mieghem 1995, Mandelbaum and Stolyar 2004, Huang et al. 2015).

In multi-server systems, dynamic scheduling of multiple classes with customer abandonment has been studied under different operational regimes. For example, Harrison and Zeevi (2004) and Atar et al. (2004) examined scheduling under critically loaded conditions. Atar et al. (2010) extended these ideas to the many-server heavy-traffic regime, proving the asymptotic optimality of the $c\mu/\theta$ rule for systems with abandonment. More recent contributions, such as Long et al. (2020), generalized these scheduling rules to incorporate non-linear cost functions and diverse patience distributions. Puha and Ward (2019) provided a comprehensive tutorial on scheduling policies for overloaded many-server queues with impatient customers. Additional recent developments include scheduling customers with heterogeneous resource requirements (Zychlinski et al. 2023), proactive service scheduling (Hu et al. 2022), hybrid healthcare systems (Zychlinski 2024), AI-assisted medical diagnostics (Cai and Zychlinski 2025), as well as group and individual therapy following mass trauma events (Zychlinski 2025).

More broadly, deterministic fluid models have been widely used to approximate the behavior of many-server queueing systems with time-varying inputs, customer abandonment, and general service and patience time distributions. These models offer tractable first-order approximations of system dynamics in overloaded regimes and have laid the foundation for performance analysis and algorithmic design in large-scale service environments (Whitt 2006a, Liu and Whitt 2011, 2012, 2014).

In this work, we show that in multi-class, multi-server systems with SL constraints, the optimal policy jointly coordinates scheduling decisions with both immediate and delayed rejections. We characterize the structure of such policies that balance cost efficiency and SL compliance.

*Queueing Systems with Rejection.* The concept of immediate customer rejection in queueing systems due to finite capacity was first introduced by Erlang (1909), who studied blocking in telephone networks. This idea was further expanded in the context of loss networks by Palm (1943). In a comprehensive review, Stidham (1985) explored methods for controlling congestion in queueing systems by restricting arrivals, focusing on the distinction between socially optimal and individually optimal controls through both static and dynamic admission policies.

In the classical Erlang-B queueing system, customers are blocked or rejected if no capacity is available. However, rejection may also occur even when waiting space is available, when long wait

times make timely service impossible. In this context, Perry and Asmussen (1995) analyzed an M/G/1 queue where customers may be partially or fully rejected based on their impatience and the system's state, deriving the steady-state distribution and exploring the busy period under various conditions. In an M/M/1 queueing system, Lewis (2001) studied a model in which rejections may occur either at arrival or immediately before service, and showed that optimal switching-curve policies exist, with average rewards increasing when pre-service rejections are more likely to be implemented. We extend this line of work by allowing delayed rejections to occur at any point during the waiting period, not just prior to service, by optimizing multi-class scheduling decisions alongside the two rejection mechanisms, and by considering a multi-server queueing environment.

Queueing systems with rejection have broad applications, including in healthcare, telecommunications, manufacturing, and cloud computing. For instance, in healthcare, rejection is sometimes necessary when timely treatment cannot be guaranteed, as discussed by Allon et al. (2013), who examined ambulance diversion in emergency departments. In telephone call centers, Gans et al. (2003) reviewed staffing rules in an Erlang-B queueing systems, in different operational regimes.

More recently, Chydzinski (2023) studied a queueing system with immediate probabilistic job rejection based on system occupancy, deriving formulas for transient and stationary throughput under various conditions, including different rejection probabilities, system loads, and inter-arrival distributions. Legros (2020) studied a multi-server queueing system with delayed rejections, focusing on optimizing time-based threshold policies to balance rejection rates and service-level objectives such as waiting time percentiles and abandonment rates. Their analysis revealed that optimal rejection thresholds can be derived from the system state, particularly the waiting time of the head-of-line customer. In contrast, we establish that combining immediate and delayed rejection mechanisms within a unified scheduling framework leads to structured, index-based policies that satisfy SL constraints while maintaining tractability.

*Queueing Systems with SL Requirements.* Service Level Agreements (SLAs) are formal commitments between a company and its customers, specifying the expected level of service, including response times and performance standards that the service provider must meet. These agreements are crucial for building trust, maintaining high service quality, and fostering long-term relationships with customers (Trienekens et al. 2004).

Most of the research regarding SL requirements has focused on staffing problems. Baron and Milner (2009), for example, introduced a period-based SLA framework for outsourced call centers, emphasizing short-term performance metrics such as rush hour delays and abandonment rates. These metrics were approximated and utilized to optimize staffing decisions, ultimately aiming to maximize profit.

Sun and Whitt (2018) studied the joint staffing and scheduling policies aimed at meeting class-specific delay targets in large-scale service systems operating under heavy traffic. They developed an effective blind, model-free head-of-line delay-ratio (HLDR) rule that stabilizes performance at class-dependent delay thresholds in a multi-class, time-varying arrival setting. Similarly, Liu et al. (2022) address differentiated service requirements by jointly designing staffing and delay-based scheduling schemes that ensure class-specific waiting time thresholds are met with high probability. Their policy achieves state-space collapse and is shown to be asymptotically valid in heavy traffic, while also performing well in simulations of moderate-sized systems.

More recently, Liu et al. (2024) addressed SL computation in a time-varying mixed-preemptive priorities queueing system, which features fluctuating arrival rates and multiple customer classes with varying priority rules. Their research provides exact and approximate methods for SL computation and applies these to an emergency department staffing problem, where service levels for different patient classes are treated as constraints. The study's computational tests, using real hospital data, demonstrate that the proposed staffing solutions effectively guarantee patients' SLs while reducing physicians' working hours.

In the current paper, we characterize the optimal scheduling policy under class-specific SL constraints and show that service-level requirements alter the structure of the optimal control: strict priority is relaxed, and delayed rejection emerges as a key mechanism for achieving feasibility at minimal cost.

## 3. The Model

We consider a Markovian $N$-server queuing model and $I$ classes of customers, $\mathcal{I} = \{1, 2, \ldots, I\}$ who arrive to the system according to a time-homogeneous Poisson process with rate $\lambda_i$, $i \in \mathcal{I}$. Service and patience times of each class are exponential with rates $\mu_i$ and $\theta_i$, $i \in \mathcal{I}$, respectively. Figure 1 illustrates our model.

Let $X_i(t)$ and $Q_i(t)$, $i \in \mathcal{I}$, denote the number of Class $i$ customers in the system and in the queue, respectively, at time $t$, $t \geq 0$. A scheduling policy $\pi$ determines the allocation of servers to customers. We consider Markovian non-anticipating policies; that is, server allocations are based on the current state $(X; Q)$ only. Under these scheduling policies, $\{(X(t); Q(t)) : t \geq 0\}$ is a Markov process. Let $Z_i(t)$ denote the number of servers occupied with Class $i$ customers at time $t$.

In addition to scheduling, we incorporate two operational rejection mechanisms. The first is *immediate rejection*, in which an arriving Class $i$ customer may be turned away upon arrival with probability $P_i$. The rejection incurs a cost of $r_i$, and the customer exits the system immediately without generating further costs.

The second mechanism is *delayed rejection*, which allows the system operator to remove a waiting customer if they remain in the queue for too long. In practice, this corresponds to a controlled

waiting-time–out mechanism applied to customers who have not yet entered service. We model this by assigning each admitted Class $i$ customer an independent exponential clock with rate $\hat{\theta}_i$, a controllable parameter. If the customer has not entered service by the time this clock expires, the customer is removed from the system and incurs a delayed-rejection cost $\hat{\alpha}_i$. Thus, the expected waiting-time threshold before a Class $i$ customer is rejected is $1/\hat{\theta}_i$.

From an implementation standpoint, the mechanism is simple: the time-out is drawn once upon admission and requires no further state monitoring. This time-based removal rule can be implemented automatically using standard queue-management functionality and is introduced here to ensure analytical tractability.

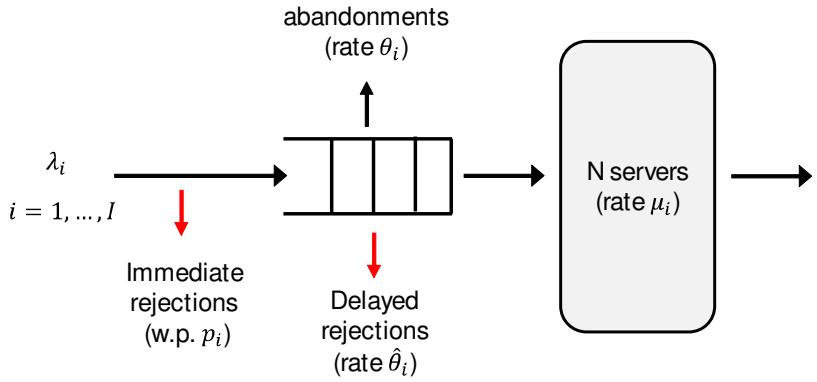


**Figure 1** **An illustration of the model with $I$ customer classes. The decision variables include the immediate rejection probabilities, the delayed rejection rates for each class, and the scheduling policy.**

Each class incurs a holding cost of $h_i$, $i \in \mathcal{I}$, per customer per unit of time. We incur an abandonment cost $\alpha_i$, $i \in \mathcal{I}$ for each Class $i$ customer who abandons while waiting in the queue. Let $R_i(t)$ and $\Gamma_i(t)$ represent the cumulative number of Class $i$ abandonments, and delayed rejections, respectively, by time $t$. Additionally, define $A_i^P(t)$ as the cumulative number of immediate rejected Class $i$ customers up to time $t$.

The aggregated cost up to time $T$ is, therefore,

$$\mathbb{E}\left[\int_0^T \left(\sum_{i\in\mathcal{I}} h_i Q_i(t)\right) \mathrm{d}t + \sum_{i\in\mathcal{I}} \left[r_i A_i^P(T) + \alpha_i R_i(T) + \hat{\alpha}_i \Gamma_i(T)\right]\right],$$

where under the Markovian modeling assumption,

$$\mathbb{E}\left[R_i(T)\right] = \theta_i \mathbb{E}\left[\int_0^T Q_i(t)\mathrm{d}t\right], \quad \mathbb{E}\left[\Gamma_i(T)\right] = \hat{\theta}_i \mathbb{E}\left[\int_0^T Q_i(t)\mathrm{d}t\right].$$

We, therefore, get that

$$\mathbb{E}\left[\int_0^T \left(\sum_{i\in\mathcal{I}} \left[\left(h_i + \alpha_i\theta_i\right) Q_i(t) + \hat{\alpha}_i\hat{\theta}_i Q_i(t)\right]\right) \mathrm{d}t + \sum_{i\in\mathcal{I}} r_i A_i^P(T)\right],$$

For simplicity of notation we introduce the "generalized" holding costs $c_i = h_i + \alpha_i \theta_i$, $i \in \mathcal{I}$. Our goal is to find a scheduling policy $\pi$ that minimizes the total expected long-run average loss, specifically:

$$\begin{aligned} \min_{\pi \in \Omega, P, \hat{\theta}} \limsup_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \int_0^T \left( \sum_{i \in \mathcal{I}} \left[ c_i Q_i(t) + \hat{\alpha}_i \hat{\theta}_i Q_i(t) \right] \right) \mathrm{d}t + \sum_{i \in \mathcal{I}} r_i A_i^P(T) \right] \\ \text{s.t.} \quad \sum_{i \in \mathcal{I}} Z_i(t) \le N; \\ 0 \le Z_i(t) \le X_i(t), \ i \in \mathcal{I}, \ \forall t \ge 0, \end{aligned} \tag{1}$$

where $\Omega$ denotes the set of admissible controls.

Problem (1) is a Markov Decision Process (MDP). However, the curse of dimensionality – stemming from the need to manage a large (or potentially infinite) state and policy space – complicates the identification and characterization of the optimal scheduling policy (Papadimitriou and Tsitsiklis 1999).

To gain structural insights into the optimal policy, we adopt a deterministic fluid model. Fluid models are widely used in service operations management to capture the first-order behavior of stochastic systems, typically arising as asymptotic limits under the Functional Law of Large Numbers (Whitt 2002, Zychlinski 2023).

The fluid models in this paper are developed in the many-server heavy-traffic regime, in which both arrival rates and server capacity scale proportionally. While the original MDP formulation can be solved numerically for very small systems, it quickly becomes computationally intractable as the number of servers or customer classes grows, and even when feasible often yields policies with limited structural transparency. In contrast, the fluid model leads to simple, index-based policies that are interpretable, easy to implement, and effective even in systems of moderate size. Such fluid approximations are known to be accurate for medium- to large-scale systems. Our numerical experiments demonstrate the accuracy and effectiveness of the developed fluid models and the derived policies in such settings.

## 4. The Fluid Model

In the fluid model, deterministic continuous rates replace stochastic processes. We use lowercase $\bar{x}_i, \bar{q}_i$ and $\bar{z}_i$, $i \in \mathcal{I}$, to denote the *average* fluid content in the system, the queue length, and the *fraction* of the service capacity assigned to Class $i$, respectively. The fraction of Class $i$ customers who are immediately rejected is $p_i$.

The above quantities satisfy the following relations for each $i \in \mathcal{I}$:

$$\begin{cases} \lambda_i (1 - p_i) = \mu_i \bar{z}_i + \left( \theta_i + \hat{\theta}_i \right) \bar{q}_i; \\ \bar{q}_i, \bar{z}_i \ge 0. \end{cases}$$

We define the fluid analogue problem to the MDP in (1) as

$$\begin{aligned}\min_{\bar{q},\bar{z},p,\hat{\theta}} & \sum_{i\in\mathcal{I}}\left[\left(c_i+\hat{\alpha}_i\hat{\theta}_i\right)\bar{q}_i+r_i\lambda_i p_i\right]\\ \text{s.t.}\quad & \lambda_i\left(1-p_i\right)=\mu_i\bar{z}_i+\left(\theta_i+\hat{\theta}_i\right)\bar{q}_i,\\ & \sum_{i\in\mathcal{I}}\bar{z}_i\le 1,\quad \bar{q}_i,\bar{z}_i\ge 0,\quad i\in\mathcal{I}.\end{aligned}$$

Noting that the first constraint is equivalent to

$$\bar{q}_i=\frac{\lambda_i(1-p_i)-\mu_i\bar{z}_i}{\theta_i+\hat{\theta}_i},\tag{2}$$

and substituting it into the objective function yields the following

$$\min_{\bar{q},\bar{z},p,\hat{\theta}}\sum_{j=1}^{\mathcal{I}}\left[\frac{\lambda_i(1-p_i)\left(c_i+\hat{\alpha}_i\hat{\theta}_i\right)}{\theta_i+\hat{\theta}_i}-\frac{\left(c_i+\hat{\alpha}_i\hat{\theta}_i\right)\mu_i}{\theta_i+\hat{\theta}_i}\bar{z}_i+r_i\lambda_i p_i\right],$$

which can be rewritten as

$$\min_{\bar{q},\bar{z},p,\hat{\theta}}\sum_{i\in\mathcal{I}}\left[\frac{\lambda_i(c_i+\hat{\alpha}_i\hat{\theta}_i)}{\theta_i+\hat{\theta}_i}-\lambda_i\left(\frac{(c_i+\hat{\alpha}_i\hat{\theta}_i)}{\theta_i+\hat{\theta}_i}-r_i\right)p_i-\frac{(c_i+\hat{\alpha}_i\hat{\theta}_i)\mu_i}{\theta_i+\hat{\theta}_i}\bar{z}_i\right].$$

Defining

$$\hat{c}_i:=\frac{c_i+\hat{\alpha}_i\hat{\theta}_i}{\theta_i+\hat{\theta}_i},\tag{3}$$

we get the following optimization problem:

$$\begin{aligned}\max_{\bar{z},p,\hat{\theta}} & \sum_{i=1}^{\mathcal{I}}\left[\hat{c}_i\mu_i\bar{z}_i+\lambda_i\left(\left(\hat{c}_i-r_i\right)p_i-\hat{c}_i\right)\right]\\ \text{s.t.}\quad & \sum_{i\in\mathcal{I}}\bar{z}_i\le 1,\\ & 0\le\bar{z}_i\le\lambda_i\left(1-p_i\right)/\mu_i,\quad p_i\in[0,1].\end{aligned}\tag{4}$$

### 4.1. Optimal Fluid Solution

We begin by characterizing the optimal value of $\hat{c}$, as defined in Equation (3).

**Lemma 1** *Under the optimal solution of (4), for any class $i\in\mathcal{I}$, the following holds:*

$$\hat{c}_i^*=\inf_{\hat{\theta}_i}\hat{c}_i=\begin{cases}c_i/\theta_i, & \text{if } c_i/\theta_i\le\hat{\alpha}_i,\\ \hat{\alpha}_i, & \text{otherwise.}\end{cases}$$

*where $\hat{\theta}_i^*=0$ or tends to $\infty$, respectively.*

In other words, Lemma 1 establishes that $\hat{c}_i^* = \min\{c_i/\theta_i, \hat{\alpha}_i\}$.

Moreover, Lemma 1 shows that the optimal delayed rejection rate, $\hat{\theta}_i$, can only take extreme values—either 0 or $\infty$. When $\hat{\theta}_i = 0$, the delayed rejection mechanism is effectively disabled, and no delayed rejections occur. Conversely, as $\hat{\theta}_i \to \infty$, customers are rejected immediately upon arrival *if they cannot be served without delay*. In this sense, the immediate rejection mechanism subsumes the delayed rejection mechanism. However, this equivalence breaks down once class-specific SL requirements are introduced, as discussed in Section 5.

Then, we proceed by defining the following index for each customer Class $i \in \mathcal{I}$:

$$\mathcal{L}_i := \left( r_i \wedge \frac{c_i}{\theta_i} \wedge \hat{\alpha}_i \right), \tag{5}$$

where $x \wedge y = \min(x, y)$.

Next, we move to the following definition and outline some useful properties of the optimal solution.

**Definition 1 (Immediately Rejectable and Non-Rejectable Classes)** *Class $i$ is said to be immediately rejectable if its rejection cost satisfies $\mathcal{L}_i = r_i$, and non-immediately rejectable otherwise.*

**Proposition 1** *Under the optimal solution of (4):*

- *For any immediately rejectable class $i \in \mathcal{I}$, the optimal queue length is $\bar{q}_i^* = 0$.*
- *For any non-immediately rejectable class $i \in \mathcal{I}$ optimal immediate rejection probability $p_i^* = 0$.*

Next, we define the $\mathcal{L}\mu$ index rule as follows:

**Definition 2 (the $\mathcal{L}\mu$ rule)** *Assign priority and service capacity to Class $i$, $i \in \mathcal{I}$, having the higher $\mathcal{L}_i\mu_i$ index.*

Theorem 1 establishes the optimality of the $\mathcal{L}\mu$ rule as the solution to (4).

**Theorem 1 (optimality of the $\mathcal{L}\mu$ rule)** *For the optimization problem (4), the $\mathcal{L}\mu$ rule is optimal. The optimal immediate rejection probability is*

$$p_i^* = \mathbf{1}_{\left\{r_i \le \left(\frac{c_i}{\theta_i} \wedge \hat{\alpha}_i\right)\right\}} \left(1 - \frac{\mu_i}{\lambda_i} \bar{z}_i\right) \tag{6}$$

*where $1_{\{x\}}$ is an indicator function that equals 1 when the condition $x$ is true, and 0 otherwise.*

We now highlight three special cases that simplify the structure of the optimal solution by restricting the rejection mechanisms, while retaining the underlying scheduling problem. The simplest way to implement these restrictions is by assigning an infinitely large cost to the mechanism we wish to neutralize.

- **Only immediate rejections are available.** In this case, delayed rejection is disabled ($\hat{\theta}_i \equiv 0$), and the cost coefficient in the $\mathcal{L}\mu$ rule, as well as the optimal immediate rejection fraction, reduce to

$$\mathcal{L}_i := \left(r_i \wedge \frac{c_i}{\theta_i}\right), \qquad p_i^* = \mathbf{1}_{\left\{r_i \le \frac{c_i}{\theta_i}\right\}} \left(1 - \frac{\mu_i}{\lambda_i}\bar{z}_i\right).$$

In particular, if $r_i > c_i/\theta_i$, then it is never optimal to reject Class $i$ upon arrival, and the policy reduces to the classical $c\mu/\theta$ rule of Atar et al. (2010). If, however, $r_i \le c_i/\theta_i$, then the index reduces to $\mathcal{L}_i\mu_i = r_i\mu_i$, and admission is reduced to match the available service capacity.

- **Only delayed rejections are available.** In this case, immediate rejection is disabled ($p_i \equiv 0$), and the cost coefficient in the $\mathcal{L}\mu$ rule reduces to

$$\mathcal{L}_i := \left(\frac{c_i}{\theta_i} \wedge \hat{\alpha}_i\right),$$

where the optimal delayed rejection rate $\hat{\theta}_i^*$ is characterized in Lemma 1. In particular, if $\hat{\alpha}_i > c_i/\theta_i$, then it is never optimal to reject Class $i$ while waiting, and the policy reduces to the classical $c\mu/\theta$ rule of Atar et al. (2010). If, however, $\hat{\alpha}_i \le c_i/\theta_i$, then the index reduces to $\mathcal{L}_i\mu_i = \hat{\alpha}_i\mu_i$.

- **No rejection is available.** If both immediate and delayed rejections are ruled out (e.g., by setting $r_i$ and $\hat{\alpha}_i$ to be larger than $c_i/\theta_i$), then Theorem 1 reduces to the classical $c\mu/\theta$ rule of Atar et al. (2010).

The implications of the $\mathcal{L}\mu$ rule differ between immediately rejectable and non-rejectable classes. For immediately rejectable classes, all customers who cannot be served upon arrival are rejected. In contrast, for non-immediately rejectable classes, no customers are rejected. This distinction follows from Proposition 1, which shows that for non-immediately rejectable classes, the optimal delayed rejection rate $\hat{\theta}$ is either zero (indicating no delayed rejections) or tends to infinity (implying immediate rejections upon admission).

When all $\mathcal{L}\mu$ indices are distinct, due to the strict priority rule, at most one class – the one that cannot be fully served – can have an optimal immediate rejection probability strictly between 0 and 1. This is because, under the optimal policy, classes with higher indices are fully admitted ($p_i = 0$), while those with lower indices are entirely rejected upon arrival ($p_i = 1$).

These results offer an important operational insight. Recall that $\mathcal{L}_i = \left(r_i \wedge \frac{c_i}{\theta_i} \wedge \hat{\alpha}_i\right)$. When $\mathcal{L}_i = r_i$, the optimal policy induces a behavior analogous to an Erlang-B queue for Class $i$: customers who cannot be immediately served are rejected or blocked without waiting. In contrast, when $\mathcal{L}_i = \left(\frac{c_i}{\theta_i} \wedge \hat{\alpha}_i\right)$, the system behaves more like an Erlang-A queue, where Class $i$ customers may wait in queue and are either eventually served, rejected, or abandon. In other words, each customer class may effectively follow a different queueing regime under the optimal policy.

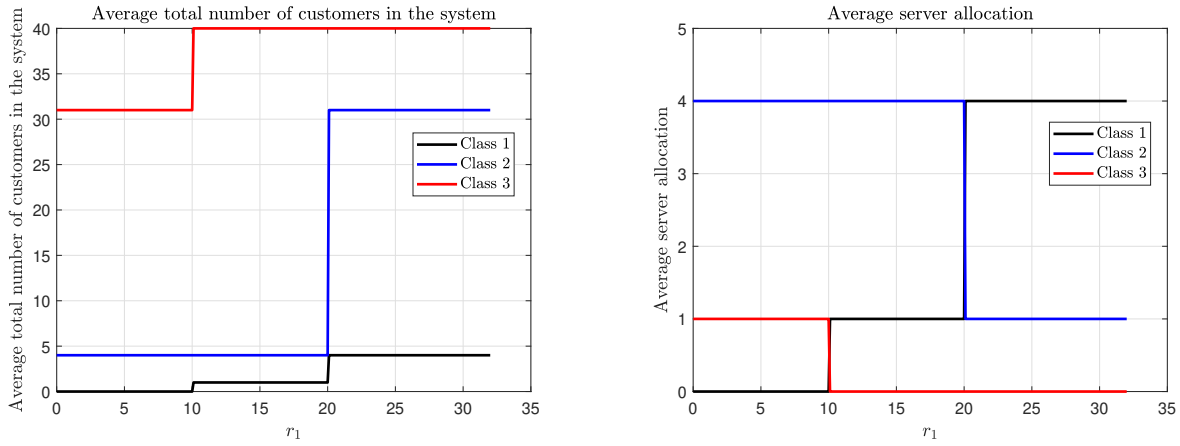


**Figure 2** **Average number of customers in the system and server allocation for each class under the $\mathcal{L}\mu$ rule. The parameters are:** $N = 5$, $\lambda = (4, 4, 4)$, $\mu = (1, 1, 1)$, $\theta = (0.1, 0.1, 0.1)$, $h = (2.8, 1.8, 0.8)$, $r = (r_1, 30, 30)$, $\alpha = (2, 2, 2)$, $\hat{\alpha} = (50, 50, 50)$.

### 4.2. A Numerical Example

To illustrate the $\mathcal{L}\mu$ policy, we consider a three-class system with five servers. Figure 2 displays the average number of each customer class in the system and the average server allocation to each class as a function of Class 1's immediate rejection cost, $r_1$. There are two priority switches, occurring at $r_1 = 10$ and $r_1 = 20$, resulting in three distinct regions. This arises because $\mathcal{L}_3 = c_3/\theta_3 = 10$, $\mathcal{L}_2 = c_2/\theta_2 = 20$, and $\mu_1 = \mu_2 = \mu_3 = 1$.

In the rightmost region, Class 1's immediate rejection cost is sufficiently high, granting it the highest priority. As a result, no immediate rejections occur, and Class 1 is fully served. In the middle region, priority shifts between Class 1 and Class 2, with the majority of servers allocated to Class 2. In this regime, approximately 75% of arriving Class 1 customers are immediately rejected. In the leftmost region, where the immediate rejection cost for Class 1 is very low, priority shifts between Class 1 and Class 3, resulting in complete immediate rejection of Class 1 customers.

### 4.3. Translation Back to the Stochastic System

The $\mathcal{L}\mu$ policy is derived through a fluid approximation. Our objective, however, is to implement this policy and evaluate its performance in the original stochastic system. Because the policy combines scheduling with both immediate and delayed rejection mechanisms, translating the fluid solution back into a stochastic setting is nontrivial. Accordingly, this section has two main objectives: first, to develop an effective and implementable translation of the policy, and second, to compare the performance of the proposed $\mathcal{L}\mu$ policy with benchmark policies.

Customer scheduling is governed by a strict priority rule. Under this rule, admission decisions and service allocation are tightly coupled, which naturally induces an immediate-rejection mechanism

when capacity is unavailable. While several implementations of immediate rejection are possible, we adopt the one that most closely mirrors the structure of the fluid-optimal policy. Specifically, for any Class $i$ with $p_i^* > 0$, an arriving customer is admitted only if a server is available upon arrival; otherwise, the customer is rejected. This rule is used to implement immediate rejections in the simulation model. Note that the rejection probabilities $p_i^*$ are not implemented through explicit randomization at arrivals. Rather, they characterize the long-run fraction of rejected Class $i$ customers in the fluid-optimal solution. The state-based admission rule described above induces rejection behavior consistent with these fractions in steady state, while remaining deterministic and threshold-like, in line with standard admission control mechanisms used in practice.

Delayed rejections are straightforward to simulate, as they closely resemble standard abandonment behavior. Each admitted Class $i$ customer is assigned two independent exponential clocks: one with rate $\theta_i$, governing endogenous abandonment, and another with rate $\hat{\theta}_i^*$, governing delayed rejection. The first clock to expire determines the outcome. If the abandonment clock expires first, a cost of $\alpha_i$ is incurred; if the delayed-rejection clock expires first, a cost of $\hat{\alpha}_i$ is incurred.

Operationally, delayed rejection is implemented by assigning each admitted customer a class-specific exponential time-out upon joining the queue. If the customer is not served before this time-out expires, the customer is removed from the system and incurs a delayed-rejection cost.

To evaluate the effectiveness of the proposed $\mathcal{L}\mu$ rule, we conduct stochastic simulations using a three-class system. Figure 3 compares the performance of our policy to two benchmarks. The first is the $c\mu/\theta$ rule without rejections, as analyzed in previous sections. The second benchmark is a dynamic threshold policy, where arriving customers are immediately rejected if the queue length exceeds a predefined threshold.

For the threshold-based policy, we explored various configurations, including both class-specific and total queue length thresholds. The results presented correspond to the best-performing configuration—a joint threshold on the total queue length, set to 10 in the left plot and 20 in the right plot of Figure 3. Specifically, once the total queue length exceeds this threshold, arriving Class 1 customers are rejected, provided that $r_1 \leq c_1/\theta_1$. In this example, only Class 1 customers are immediate rejectable.

The figure reports the long-run average system cost under each policy for different values of the immediate rejection cost, $r_1$. We use the same system parameters as in Figure 2.

The simulation results reveal several key insights:

- The $\mathcal{L}\mu$ rule consistently outperforms the $c\mu/\theta$ rule, except when $r_1$ is sufficiently large, in which case both policies converge to similar outcomes, as immediate rejections become suboptimal.
- The dynamic threshold policy performs better than the $c\mu/\theta$ rule when $r_1$ is small, but its performance deteriorates for larger $r_1$, as it continues to reject customers even when rejection costs are prohibitively high.

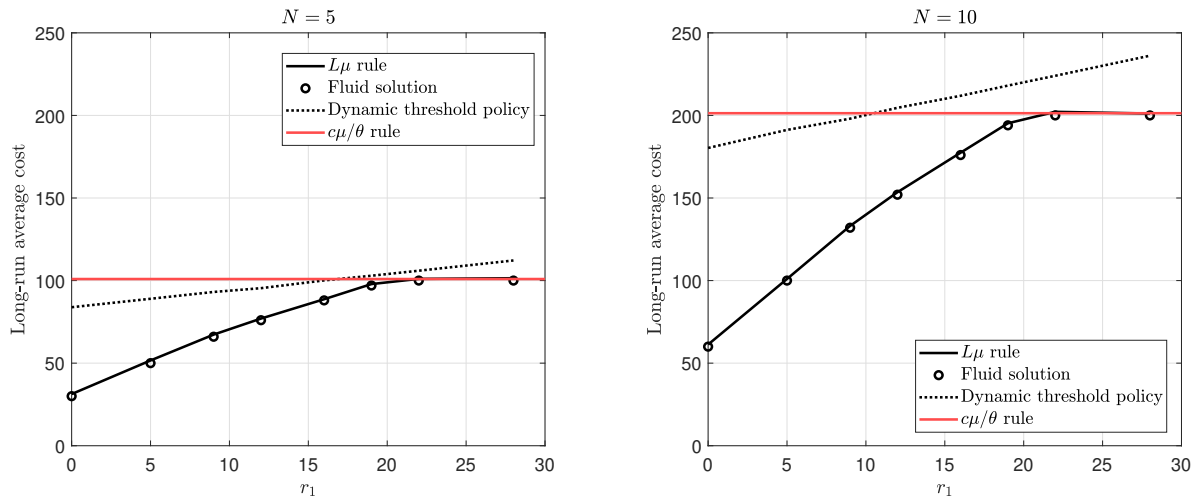


**Figure 3** **Cost comparison between the $\mathcal{L}\mu$ implementation in the stochastic system, the fluid model, and the $c\mu/\theta$ rule for varying rejection costs. The parameters are:** $\mu = (1,1,1)$, $\theta = (0.1,0.1,0.1)$, $h = (2.8,1.8,0.8)$, $r = (r_1,30,30)$, $\alpha = (2,2,2)$, $\hat{\alpha} = (50,50,50)$**. In the left plot,** $N = 5$**,** $\lambda = (4,4,4)$**; in the right plot** $N = 10$**,** $\lambda = (8,8,8)$**.**

- In contrast, the $\mathcal{L}\mu$ rule inherently adjusts to system conditions by dynamically determining the optimal rejection probability $p_1^*$, thus avoiding excessive rejection costs without the need to fine-tune external thresholds.

Moreover, the $\mathcal{L}\mu$ policy provides structural advantages: it offers an analytical characterization and requires no trial-and-error calibration of thresholds for each operational scenario. Notably, these performance advantages persist regardless of system scale. Overall, the $\mathcal{L}\mu$ rule performs nearly as well as the fluid-optimal solution, demonstrating both practical effectiveness and robust theoretical grounding.

## 5. Integrating Class-Specific Service Level (SL) Requirements

In many service systems, maintaining a high standard of service quality is essential. To achieve this, SL requirements are often established for each customer class. The SL requirements considered in this paper impose an upper limit on the average waiting time that customers in each class can experience before receiving service. By setting these SL thresholds, service providers can ensure timely service and prevent excessive waiting times. In this section, we incorporate class-specific SL requirements into the scheduling problem, accounting for both rejection types.

Let $\bar{w}_i$ denote average waiting time of Class $i$ customers, and $\tau_i \geq 0$, $i \in \mathcal{I}$ denote the maximal average waiting time allowed for Class $i$ customers. We, therefore, require that

$$\bar{w}_i \leq \tau_i, \quad i \in \mathcal{I}.$$

Applying Little's law, we get the following constraint:

$$\bar{w}_i = \frac{\bar{q}_i}{\lambda_i (1-p_i)} \leq \tau_i.$$

Then, plugging in the steady-state equation in (2), yields the following equivalent

$$\frac{\lambda_i(1-p_i) - \mu_i \bar{z}_i}{\lambda_i (1-p_i) \left(\theta_i + \hat{\theta}_i\right)} \leq \tau_i,$$

which can be rewritten as

$$\bar{z}_i \geq \frac{\lambda_i}{\mu_i}(1-p_i)\left(1 - (\theta_i + \hat{\theta}_i)\tau_i\right). \tag{7}$$

Trivially, if the solution to the problem without SL constraints already satisfies the SL constraints, then this solution remains optimal. Note also that per Proposition 1, in case $\mathcal{L}_i = r_i$ or $\mathcal{L}_i = \hat{\alpha}_i$, we have $\bar{q}_i = 0$, so the constraint is satisfied. Therefore, only when $\mathcal{L} = c_i/\theta_i$ (implying that optimally $\hat{\theta}_i = 0$ and $p_i = 0$), the SL constraint may be violated. We, therefore, introduce a specific subset of customer classes – the constraint-breaching Classes.

**Definition 3 (a constraint-breaching (CB) class)** *A non-rejectable Class $i \in \mathcal{I}$ having $\mathcal{L}_i = c_i/\theta_i$ is called a CB class, if $\tau_i < 1/\theta_i$. Let $\mathcal{I}^{CB} \subset \mathcal{I}$ denote the subset of all CB classes.*

In other words, a CB class is one where the average patience time is longer than the maximum allowable waiting time for this class, indicating that if all customers were allowed to wait according to their patience, the average waiting time would exceed the specified requirement. Our optimization problem can be written as follows:

$$\begin{aligned}
\max_{\bar{z},p,\hat{\theta}} \; & \sum_{i=1}^{\mathcal{I}} \left[\hat{c}_i \mu_i \bar{z}_i + \lambda_i \left(\left(\hat{c}_i - r_i\right) p_i - \hat{c}_i\right)\right] \\
\text{s.t.} \; & \sum_{i \in \mathcal{I}} \bar{z}_i \leq 1, \\
& 0 \leq \bar{z}_i \leq \lambda_i \left(1-p_i\right)/\mu_i, \; i \in \mathcal{I} \\
& \lambda_i (1-p_i)\left(1 - (\theta_i + \hat{\theta}_i)\tau_i\right)/\mu_i \leq \bar{z}_i, \; i \in \mathcal{I}^{CB},
\end{aligned} \tag{8}$$

where the last constraint is the SL requirement, which applies only to CB classes, since by definition it holds for all the other classes.

**Lemma 2 (Exclusive control usage for CB classes)** *In the optimal solution of (8), for any specific set of $\tau_i$, each CB class utilizes at most one rejection mechanism, either immediate of delayed.*

Next, we define the $\mathcal{L}^{\mathrm{S}}\mu$ rule, which will play a crucial role in our analysis:

**Definition 4 (The $\mathcal{L}^{\mathrm{S}}\mu$ Rule)** *Assign priority to Class $i$, $i \in \mathcal{I}$, having the higher $\mathcal{L}_i^S \mu_i$ index, where*

$$\mathcal{L}_i^S = \begin{cases} \mathcal{L}_i + 1_{\left\{\bar{z}_i \le \frac{\lambda_i}{\mu_i}(1-\theta_i\tau_i)\right\}} \left(\hat{\alpha}_i - \mathcal{L}_i \wedge \frac{r_i - \mathcal{L}_i}{1-\tau_i\theta_i}\right), & \text{if } i \in \mathcal{I}^{CB}, \\ \mathcal{L}_i\ , & \text{Otherwise}, \end{cases}$$

Notice that $\left(\hat{\alpha}_i - \mathcal{L}_i \wedge \frac{r_i - \mathcal{L}_i}{1-\tau_i\theta_i}\right)$ is positive because for CB classes $\mathcal{L}_i = c_i/\theta_i$, meaning that $\mathcal{L}_i < \hat{\alpha}_i$, $\mathcal{L}_i < r_i$, and also $\tau_i < 1/\theta_i$.

In terms of resource allocation, the $\mathcal{L}^{\mathrm{S}}\mu$ rule prescribes that resources are allocated to each CB-Class $i$ according to the adjusted index

$$\mathcal{L}_i^{\mathrm{S}} = \mathcal{L}_i + \min\left\{\hat{\alpha}_i - \mathcal{L}_i,\ \frac{r_i - \mathcal{L}_i}{1-\tau_i\theta_i}\right\},$$

until the allocated capacity for class $i$ reaches the threshold $\lambda_i(1-\theta_i\tau_i)/\mu_i$. This threshold reflects a fraction $(1-\theta_i\tau_i) \in (0,1]$ of the class's total offered load $\lambda_i/\mu_i$. Once this point is reached, the SL constraint for class $i$ is satisfied, and its index returns to the baseline value $\mathcal{L}_i^{\mathrm{S}} = \mathcal{L}_i$. Any remaining capacity is then allocated based on this baseline index.

If the available capacity is insufficient to satisfy the SL constraints, the system must use rejection mechanisms to compensate. The more cost-effective option is selected by comparing the marginal cost of the two rejection modes: delayed rejection, with cost $\hat{\alpha}_i - \mathcal{L}_i$, versus immediate rejection, with cost $\frac{r_i - \mathcal{L}_i}{1-\tau_i\theta_i}$. The system chooses the minimum of these two to maintain feasibility while minimizing cost.

Theorem 2 establishes the optimality of the $\mathcal{L}^{\mathrm{S}}\mu$ index rule when incorporating SL requirements.

**Theorem 2 (optimality under SL requirements)** *For the optimization problem (8), which incorporates SL requirements, the $\mathcal{L}^S\mu$ rule is optimal. The adjusted optimal immediate rejection probability for Class $i$ is given by:*

$$p_i^* = \begin{cases} 1_{\left\{\hat{\alpha}_i \ge \frac{r_i - \mathcal{L}_i\tau_i\theta_i}{1-\tau_i\theta_i}\right\}} \left(1 - \frac{\mu_i\bar{z}_i}{\lambda_i(1-\theta_i\tau_i)}\right), & \text{if } i \in \mathcal{I}^{CB}, \\ 1_{\left\{\frac{c_i}{\theta_i} \ge r_i,\ \hat{\alpha}_i \ge r_i\right\}} \left(1 - \frac{\mu_i}{\lambda_i}\bar{z}_i\right), & \text{otherwise.} \end{cases}$$

*The adjusted optimal delayed rejection rate for Class $i$ is:*

$$\hat{\theta}_i^* = \begin{cases} 1_{\left\{\hat{\alpha}_i < \frac{r_i - \mathcal{L}_i\tau_i\theta_i}{1-\tau_i\theta_i}\right\}} \left(\frac{1}{\tau_i} - \frac{\mu_i\bar{z}_i}{\lambda_i\tau_i} - \theta_i\right), & \text{if } i \in \mathcal{I}^{CB}, \\ \infty, & \text{if } i \notin \mathcal{I}^{CB} \text{ and } \mathcal{L}_i = \hat{\alpha}_i, \\ 0, & \text{if } i \notin \mathcal{I}^{CB} \text{ and } \mathcal{L}_i \ne \hat{\alpha}_i. \end{cases}$$

A key distinction between the basic model and the system with SL requirements concerns the optimal delayed-rejection rate $\hat{\theta}_i$. Without SL constraints, Lemma 1 shows that $\hat{\theta}_i^*$ is always pushed to the boundaries—either 0 (no delayed rejection) or $\infty$ (effectively immediate rejection).

Under SL requirements, this structure no longer holds for CB classes. When capacity allocation alone cannot satisfy the SL constraint, the system must rely on rejection mechanisms to control congestion and ensure compliance. As shown in Lemma 2, at most one rejection mechanism is used per CB class, selected by comparing the marginal costs of immediate and delayed rejections. If delayed rejection is more cost-effective, the system sets $\hat{\theta}_i^*$ to an interior value so that the expected waiting time exactly meets the SL threshold $\tau_i$. Consequently, under SL constraints, $\hat{\theta}_i^*$ may take any value in $[0, \infty)$, depending on system conditions.

The qualitative change in the behavior of $\hat{\theta}_i$ under SL constraints follows directly from the role this decision variable plays in the optimization problem. In the basic model, delayed rejection affects the objective only through cost terms in the objective function. As a result, the objective is monotone in $\hat{\theta}_i$, and the optimizer always pushes it to an extreme value—either 0 (never reject) or $\infty$ (effectively immediate rejection), as shown in Lemma 1.

Under SL constraints, $\hat{\theta}_i$ enters the problem in a fundamentally different way: it directly determines the feasible set by controlling the steady-state waiting time. In particular, for CB classes, the SL constraint fixes the admissible waiting time and thus implicitly pins down $\hat{\theta}_i$ whenever capacity allocation alone is insufficient. In this case, $\hat{\theta}_i$ is no longer chosen to minimize cost, but to satisfy the SL constraint with equality, which may require an interior value. This mechanism is illustrated in Section 5.3 and Figure 6, where delayed rejection adjusts continuously to enforce the waiting-time target.

### 5.1. Understanding the Mechanism of the $\mathcal{L}^{\mathrm{S}}\mu$ Rule

The key distinction between the $\mathcal{L}\mu$ and $\mathcal{L}^{\mathrm{S}}\mu$ rules is with regard to the CB classes. To ensure that these classes meet the SL constraints, the operator has three options: allocate additional resources, increase delayed rejections rate, or increase immediate rejection probability of arriving customers. Recall that according to Lemma 2 it is optimal to use at most one control. This trade-off is reflected in the elevated index for CB classes, which temporarily grants them higher priority when allocating resources. Once a class receives sufficient resources to satisfy the SL constraint, its index reverts to the standard $\mathcal{L}\mu$ value.

Figure 4 illustrates the fundamental difference between the $\mathcal{L}\mu$ and $\mathcal{L}^{\mathrm{S}}\mu$ rules within the fluid model. The total fluid capacity, represented by the large container in the figure, needs to be allocated among five customer classes, each depicted as a smaller container. The size of these containers reflects the capacity required by each class.

In Case **(a)**, the five classes are ordered according to their respective $\mathcal{L}\mu$ indices, with $\mathcal{L}_1\mu_1$ being the largest. The left side of the figure shows the initial stages of capacity allocation, while the right side shows the final allocation. Following the $\mathcal{L}\mu$ rule, capacity is allocated sequentially: first to Classes 1 through 3, with the remaining capacity going to Class 4.

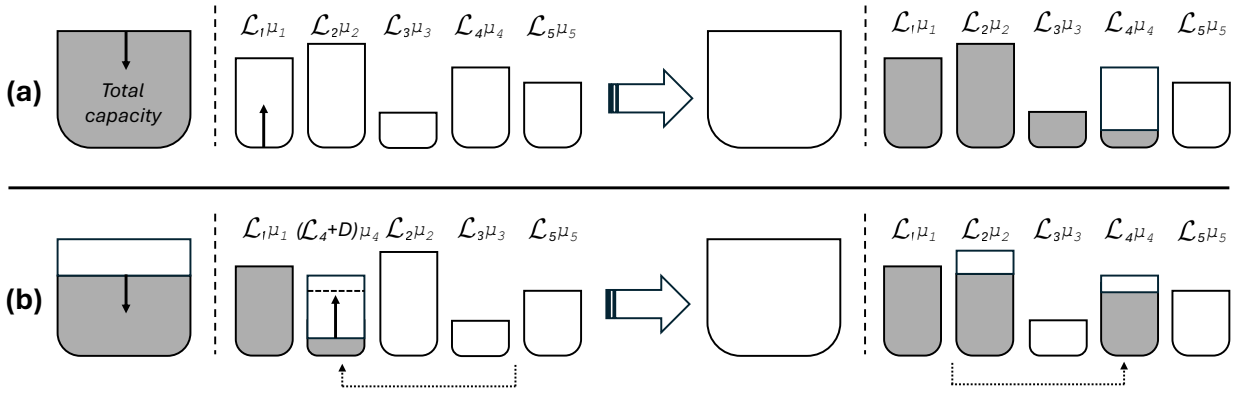


**Figure 4** **Illustration of fluid allocation according to the $\mathcal{L}\mu$ rule (Case a) and the $\mathcal{L}^{\mathrm{S}}\mu$ rule (Case b).**

Case **(b)** illustrates the allocation process under the $\mathcal{L}^{\mathrm{S}}\mu$ rule, where Class 4 is a CB class. According to the $\mathcal{L}^{\mathrm{S}}\mu$ rule, the index for Class 4 increases to $(\mathcal{L}_4 + D)\mu_4$, where $D := (r_4 - \mathcal{L}_4)/(1 - \tau_4\theta_4) \wedge \hat{\alpha}_4 - \mathcal{L}_4$, making it the second highest index after Class 1. Consequently, after allocating the necessary capacity to Class 1, capacity allocation shifts to Class 4 until the SL constraint is satisfied (indicated by the fluid reaching the dotted line within Class 4's container). At this point, Class 4's index reverts to $\mathcal{L}_4\mu_4$, and its priority drops, allowing the remaining capacity to be allocated to Class 2, leaving no capacity for Class 3.

### 5.2. Translation of the Fluid $\mathcal{L}^{\mathrm{S}}\mu$ Rule for the Stochastic System

The translation of the $\mathcal{L}^{\mathrm{S}}\mu$ rule is more complex than that of the $\mathcal{L}\mu$ rule due to the presence of CB classes and their index dependency on the allocated capacity $\bar{z}$. In general, customers with a higher $\mathcal{L}^{\mathrm{S}}\mu$ index are prioritized. For CB classes, this involves two distinct indices (see Definition 4). In the fluid model, capacity is allocated to CB classes by increasing the index until a $(1 - \theta_i\tau_i)$ share of the class's requirements is served. The remaining $\theta_i\tau_i$ share of customers is served according to the original $\mathcal{L}\mu$ index. While there may be different interpretations of this policy in the stochastic system, we propose an intuitive one: predetermined capacity is allocated to a CB Class $i$ so that the SL requirement is satisfied, i.e.,

$$\frac{\lambda_i}{\mu_i}(1 - p_i)\left(1 - (\theta_i + \hat{\theta}_i)\tau_i\right),$$

with the remaining capacity distributed among other classes according to their $\mathcal{L}\mu$ indices.

Figure 5 compares the performance of the $\mathcal{L}\mu$ rule, implemented in the stochastic system via simulation, with the fluid-optimal solution (represented by circles), across varying levels of Class 1's service level (SL) threshold, $\tau_1$.

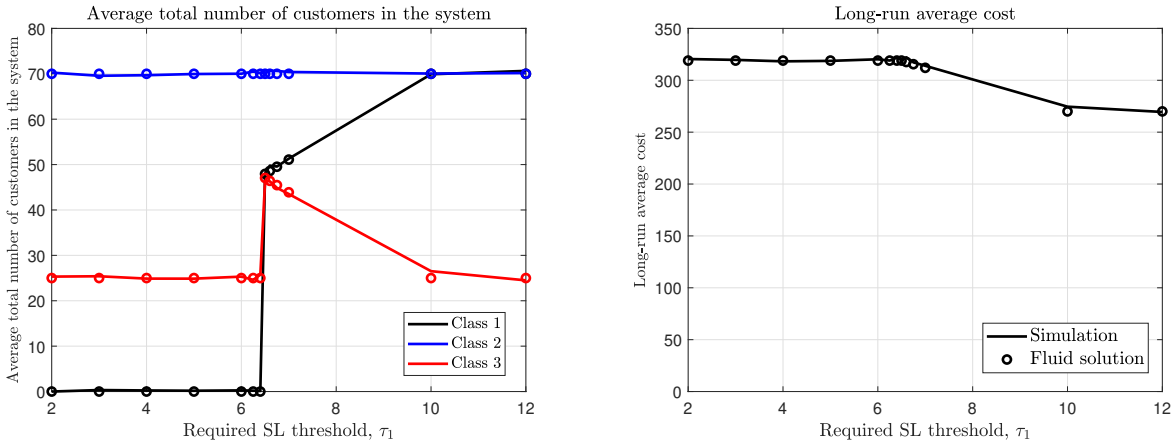


**Figure 5** **Optimal number of customers in the system and optimal cost — Simulation of the $\mathcal{L}\mu$ rule versus the fluid solution (represented by circles) for varying service level (SL) thresholds. The parameters are:** $N = 5$, $\lambda = (7,7,7)$, $\mu = (1,1,1)$, $\theta = (0.1,0.1,0.1)$, $h = (0.8,1.8,2.8)$, $r = (17,30,30)$, $\alpha = (2,2,2)$, $\hat{\alpha} = (50,50,50)$.

When $\tau_1$ is high, the system allocates most of its capacity to Class 3. As $\tau_1$ decreases, more capacity is redirected to Class 1 in order to meet its SL requirements, thereby reducing the resources available to Class 3. However, when $\tau_1$ becomes sufficiently low, Class 1 customers begin to be immediately rejected, leading to a reallocation of capacity back to Class 3.

In terms of implementation, both rejection mechanisms follow the same structure as in the basic model only with different values of $p_i^*$ and $\hat{\theta}_i^*$. For immediate rejections, a Class $i$ customer with $p_i^* > 0$ is admitted only if a server is available upon arrival; otherwise, the customer is immediately rejected.

For delayed rejections, each admitted Class $i$ customer is assigned two independent exponential clocks: one with rate $\theta_i$ is for abandonments, and another with rate $\hat{\theta}_i^*$ is for delayed rejections. The first clock to expire determines whether it is a rejection or an abandonment.

### 5.3. Impact of Service Level Requirements

We start this section by examining the effect of the SL required threshold on the optimal delayed rejection rate and long-run average cost. Figure 6 provides two examples for a three-class setting, where Class 3 is a CB class (i.e., a non-immediately rejectable class with $\mathcal{L}_3 = c_3/\theta_3$ and $\tau_3 < 1/\theta_3$). According to the basic $\mathcal{L}\mu$ indexes, Class 3 has the lowest priority among the three classes ($\mathcal{L}_1\mu_1 = 30$, $\mathcal{L}_2\mu_2 = 20$, $\mathcal{L}_3\mu_3 = 10$). However, once $\tau_3$ drops below 10, the SL requirement for Class 3 is violated.

Simply switching to the $\mathcal{L}^{\mathrm{S}}\mu$ indexes will not resolve the issue since, according to Definition 4, Class 3's index will increase to $\mathcal{L}_3^{\mathrm{S}}\mu_3 = \hat{\alpha}_3\mu_3 = 15$, but it will still remain last in prioritization. According to Theorem 2, meeting the SL requirement in this situation requires increasing the controlled abandonment rate $\hat{\theta}_3$.

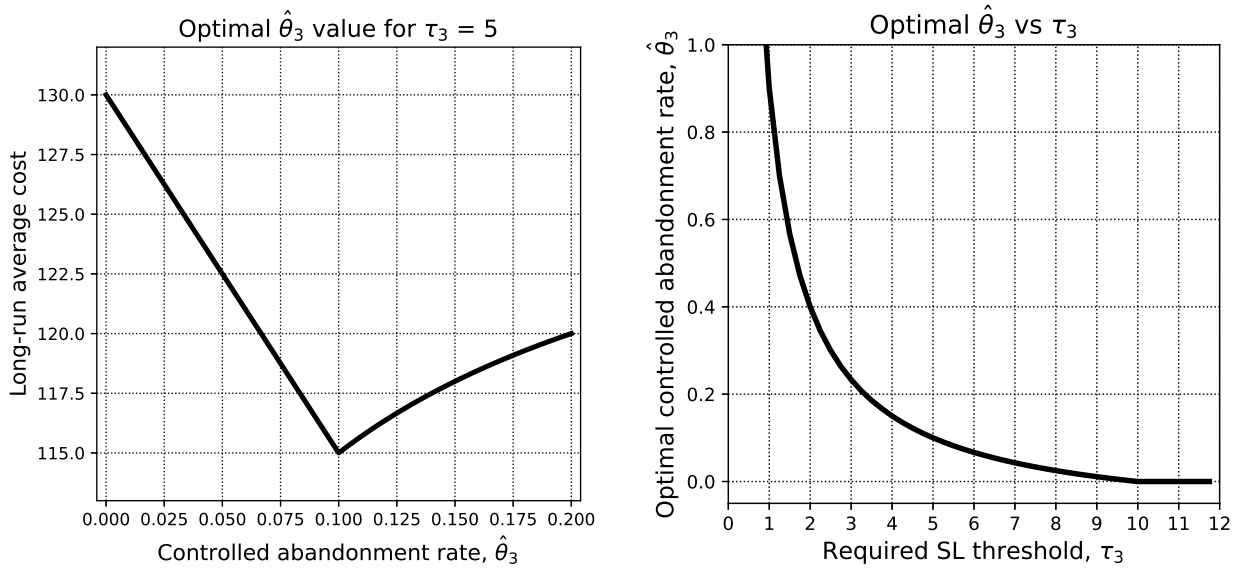


**Figure 6** **An illustration of the relationship between the SL required threshold $\tau_3$, the controlled abandonment rate $\hat{\theta}$, and the long-run average cost. The parameters are:** $N = 10,\ \lambda = (6,6,6),\ \mu = (1,1,1),\ \theta = (0.1,0.1,0.1),\ h = (2.8,1.8,0.8),\ r = (30,30,30),\ \alpha = (2,2,2),\ \hat{\alpha} = (50,50,15),\ \tau = (10,10,\tau_3)$.

Figure 6(a) shows the long-run average cost for different values of $\hat{\theta}_3$ when $\tau_3 = 5$. It is evident that $\hat{\theta}_3^* = 0.1$ achieves the lowest cost, which is indeed the global optimum. In the SL requirement (7), the right-hand side becomes negative when $\hat{\theta}_i > 1/\tau_i - \theta_i$. Since $\bar{z}_i$ is non-negative, increasing $\hat{\theta}_i$ beyond this threshold negatively impacts the objective function.

Figure 6(b) shows the dependency between the optimal delayed rejection rate and the threshold $\tau_3$. Intuitively, as the threshold increases, it becomes easier to satisfy the SL requirement allowing the controlled abandonment rate to decrease. Note that when $\tau_3 \geq 10$, Class 3 is no longer a CB class, as the SL requirement is satisfied. The optimal rate in this range is, therefore, $\hat{\theta}_3^* = 0$, which is aligned with the original $\mathcal{L}\mu$ rule.

Figure 7 presents three examples illustrating the optimal server allocation (right panels) and the corresponding long-run average cost (left panels) as functions of the SL threshold $\tau_3$.

**Example 1.** Here, Class 3 has the second-highest priority under the $\mathcal{L}^{\mathrm{S}}\mu$ rule, with $\mathcal{L}_3^{\mathrm{S}} = \hat{\alpha}_3 = 28$. We identify three regimes based on the value of $\tau_3$:

i. **High threshold** ($\tau_3 \geq 10$): Class 3 is not constrained by the SL requirement, which is satisfied without allocating any servers. In this regime, the solutions under the $\mathcal{L}^{\mathrm{S}}\mu$ and $\mathcal{L}\mu$ rules coincide.

ii. **Moderate threshold** ($\tau_3 \in \left[\frac{40}{7}, 10\right)$): To meet the SL constraint, capacity is reallocated from Class 2 to Class 3. Once the constraint is met, any remaining servers are returned to Class 2. As $\tau_3$ decreases, more capacity is allocated to Class 3 at the expense of Class 2.

iii. **Low threshold** ($\tau_3 < \frac{40}{7}$): All capacity from Class 2 has been shifted to Class 3, yet the SL constraint is still not satisfied. To compensate, the delayed rejection rate $\hat{\theta}_3$ is increased, leading

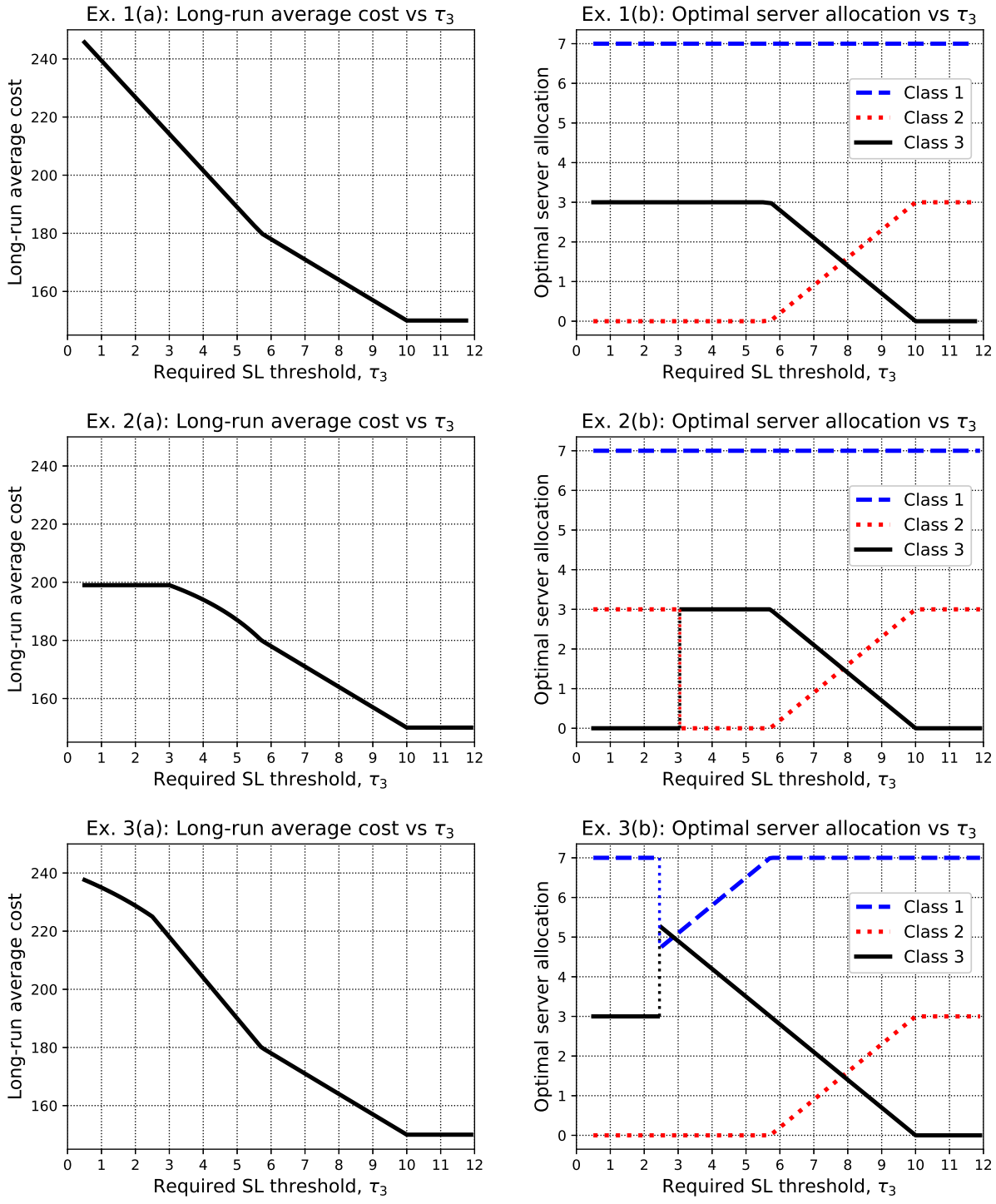


**Figure 7** **Effect of service level (SL) requirements on the optimal cost and resource allocation. Parameters:** $N = 10$, $\lambda = (7, 7, 7)$, $\mu = (1, 1, 1)$, $\theta = (0.1, 0.1, 0.1)$, $h = (2.8, 1.8, 0.8)$, $r = (30, 30, r_3)$, $\alpha = (2, 2, 2)$, $\hat{\alpha} = (50, 50, \hat{\alpha}_3)$, **and** $\tau = (10, 10, \tau_3)$. **Example 1:** $r_3 = 30$, $\hat{\alpha}_3 = 28$; **Example 2:** $r_3 = 17$, $\hat{\alpha}_3 = 50$; **Example 3:** $r_3 = 25$, $\hat{\alpha}_3 = 50$.

to a sharp rise in cost. In this regime, the allocation remains fixed (full capacity transfer from Class 2 to Class 3), but $\hat{\theta}_3$ grows as $\tau_3$ decreases.

**Example 2.** We now increase $\hat{\alpha}_3$ to 50, making delayed rejections suboptimal, and set $r_3 = 17$. Regimes (i) and (ii) remain unchanged, but regime (iii) now splits into two sub-cases:

iii.a. **Partial rejection** ($\tau_3 \in \left[3, \frac{40}{7}\right)$): After exhausting all capacity from Class 2, the system begins to immediately reject some Class 3 customers to satisfy the SL constraint. The cost increases nonlinearly due to Definition 4, where the marginal cost of rejection decreases with $\tau_3$.

iii.b. **Full rejection** ($\tau_3 < 3$): The marginal rejection cost becomes sufficiently low such that full rejection of Class 3 becomes optimal. No additional cost is incurred for further reductions in $\tau_3$.

**Example 3.** We retain a high $\hat{\alpha}_3$ and increase the immediate rejection cost to $r_3 = 25$. Again, regimes (i) and (ii) are unchanged, while regime (iii) is subdivided:

iii.a. **Reallocation from Class 1** ($\tau_3 \in \left[\frac{5}{2}, \frac{40}{7}\right)$): Since immediate rejection remains costly, the system reallocates capacity from Class 1 to Class 3. This causes a steeper linear increase in cost.

iii.b. **Threshold rejection** ($\tau_3 < \frac{5}{2}$): At this point, the marginal rejection cost becomes lower than the cost of reallocating capacity from Class 1 (though not from Class 2), prompting immediate rejection of Class 3 customers. Class 1 capacity is released, and cost again increases nonlinearly.

To conclude, in all three examples, the optimal server allocation and long-run cost coincide for $\tau_3 \geq \frac{40}{7}$. However, for $\tau_3 < \frac{40}{7}$, differences in $\hat{\alpha}_3$ and $r_3$ lead to distinct allocation strategies and cost behaviors, highlighting the sensitivity of system performance to these parameters.

## 6. Concluding Remarks and Future Research Directions

We study the optimal scheduling policy for multi-server queues with multiple customer classes, focusing on two key operational controls: immediate rejections (upon arrival) and delayed rejections (controlled abandonment during waiting times). These mechanisms allow the system operator to manage capacity and minimize long-run average cost effectively. The model is broadly applicable across various domains, including service systems, call centers, healthcare, and cloud services.

By developing a corresponding fluid model, we derive the $\mathcal{L}\mu$ rule—an index-based policy that efficiently balances holding costs, service and abandonment rates as well as immediate and delayed rejection costs. Under optimal capacity allocation, each customer class behaves either as an Erlang-B queue (when immediate rejections occur) or as an Erlang-A queue (when customers wait and may eventually be rejected via delayed rejection). Simulation experiments confirm the practical effectiveness of the $\mathcal{L}\mu$ rule even in moderate-sized systems.

When class-specific service level (SL) requirements are introduced, the $\mathcal{L}\mu$ rule is extended to the $\mathcal{L}^{\mathrm{S}}\mu$ rule, which incorporates SL constraints into the prioritization process. Under the $\mathcal{L}^{\mathrm{S}}\mu$ rule, strict

priority is relaxed: a class with a lower $\mathcal{L}\mu$ index but a tighter SL requirement may receive capacity over a higher-priority class. Translating the fluid-based solution back to the original stochastic system yields a two-stage index policy, which ensures cost-effective scheduling while meeting all SL constraints.

Looking ahead, there are several promising directions for future research. One direction involves designing policies for time-varying systems, where arrival rates and staffing levels fluctuate over time. This would require incorporating time-dependent scheduling decisions, along with adaptive immediate and delayed rejection strategies. Another extension could consider more complex queueing networks, such as tandem networks—where rejections may occur at intermediate stages—or parallel-station networks, where customers may be redirected rather than rejected. Finally, a valuable direction is to develop a decision-support tool for selecting the optimal service level threshold ($\tau$). This would involve integrating the benefit of meeting SL requirements into the cost function, allowing for analysis of the trade-off between reducing $\tau$ and the resulting operational cost.

### Declarations

*Funding.* This work was partially supported by ISF Grant 277/21, the Israel National Institute for Health Policy Research (Grant 2021/160/R), and the Bernard M. Gordon Center for Systems Engineering at the Technion.

*Competing interests.* The authors have no competing interests to declare that are relevant to the content of this article.

## References

Allon G, Deo S, Lin W (2013) The impact of size and occupancy of hospital on the extent of ambulance diversion: Theory and evidence. *Operations Research* 61(3):544–562.

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

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

Baron O, Milner J (2009) Staffing to maximize profit for call centers with alternate service-level agreements. *Operations Research* 57(3):685–700.

Bleustein C, Rothschild D, Valen A, Valatis E, Schweitzer L, Jones R (2014) Wait times, patient satisfaction scores, and the perception of care. *The American Journal of Managed Care* 20(5):393–400.

Cai J, Zychlinski N (2025) When ai is not enough: Reducing diagnostic errors with radiologist oversight. *Service Science* .

Chydzinski A (2023) Throughput of the queue with probabilistic rejections. *IEEE Access* .

Cox D, Smith W (1961) Queues. *Methuen, London* .

Erlang AK (1909) The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik* 20:33–39.

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

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

Harrison J, Zeevi A (2005) A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management* 7(1):20–36.

Hu Y, Chan C, Dong J (2022) Optimal scheduling of proactive service with customer deterioration and improvement. *Management Science* 68(4):2533–2578.

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

Jones P, Peppiatt E (1996) Managing perceptions of waiting times in service queues. *International Journal of Service Industry Management* 7(5):47–61.

Legros B (2020) Late-rejection, a strategy to perform an overflow policy. *European Journal of Operational Research* 281(1):66–76.

Lewis M (2001) Average optimal policies in a controlled queueing system with dual admission control. *Journal of Applied Probability* 38(2):369–385.

Liu R, Ouyang H, Wang C, Xie X (2024) Service-level computation in time-varying queueing system with priorities: Application to physician staffing in the emergency department. *IISE Transactions* 1–33.

Liu Y, Sun X, Hovey K (2022) Scheduling to differentiate service in a multiclass service system. *Operations Research* 70(1):527–544.

Liu Y, Whitt W (2011) A network of time-varying many-server fluid queues with customer abandonment. *Operations Research* 59(4):835–846.

Liu Y, Whitt W (2012) The $g_t/gi/s_t + gimany - serverfluidqueue. Queueing\ Systems 71:405--444$.

Liu Y, Whitt W (2014) Algorithms for time-varying networks of many-server fluid queues. *INFORMS Journal on Computing* 26(1):59–73.

Long Z, Shimkin N, Zhang H, Zhang J (2020) Dynamic scheduling of multiclass many-server queues with abandonment: The generalized c$\mu$/h rule. *Operations Research* 68(4):1218–1230.

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

Michael M, Schaffer S, Egan P, Little B, Pritchard P (2013) Improving wait times and patient satisfaction in primary care. *The Journal for Healthcare Quality (JHQ)* 35(2):50–60.

Palm C (1943) Intensitätsschwankungen im Fernsprechverkehr. *Ericsson Technics* 44:1–189.

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

Perry D, Asmussen S (1995) Rejection rules in the $M/G/1$ queue. *Queueing Systems* 19:105–130.

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

Silvester K, Lendon R, Bevan H, Steyn R, Walley P (2004) Reducing waiting times in the nhs: is lack of capacity the problem? *Clinician in Management* 12(3).

Stidham S (1985) Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control* 30(8):705–713.

Sun X, Whitt W (2018) Delay-based service differentiation with many servers and time-varying arrival rates. *Stochastic Systems* 8(3):230–263.

Trienekens J, Bouman J, Van Der Zwan M (2004) Specification of service level agreements: Problems, principles and practices. *Software Quality Journal* 12:43–57.

Van Mieghem J (1995) Dynamic scheduling with convex delay costs: The generalized c$\mu$ rule. *The Annals of Applied Probability* 809–833.

Whitt W (2002) *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and their Application to Queues* (Springer Science & Business Media).

Whitt W (2006a) Fluid models for multiserver queues with abandonments. *Operations Research* 54(1):37–54.

Whitt W (2006b) Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management* 15(1):88–102.

Worthington K (2004) Customer satisfaction in the emergency department. *Emergency Medicine Clinics* 22(1):87–102.

Zychlinski N (2023) Applications of fluid models in service operations management. *Queueing Systems* 103(1):161–185.

Zychlinski N (2024) Managing queues with reentrant customers in support of hybrid healthcare. *Stochastic Systems* 14(2):167–190.

Zychlinski N (2025) An operational view on managing mass trauma events. *Manufacturing & Service Operations Management* .

Zychlinski N, Chan C, Dong J (2023) Managing queues with different resource requirements. *Operations Research* 71(4):1387–1413.

## Appendix A: Proofs of Analytical Results

***Proof of Lemma 1:*** Let us take the following derivative:

$$(\hat{c}_i)'_{\hat{\theta}_i} = \frac{\hat{\alpha}_i(\theta_i+\hat{\theta}_i)-(c_i+\hat{\alpha}_i\hat{\theta}_i)}{\left(\theta_i+\hat{\theta}_i\right)^2} = \frac{\hat{\alpha}_i\theta_i - c_i}{\left(\theta_i+\hat{\theta}_i\right)^2}.$$

Since we are interested in comparing the derivative to zero, and the denominator is positive because we assume $\theta_i > 0$, we only need to consider the numerator. The numerator, which is independent of $\hat{\theta}_i$, is either non-negative or non-positive. With $0 \le \hat{\theta}_i < \infty$, we find that $\hat{c}_i$ always lies between $\hat{\alpha}_i$ (when $\hat{\theta}_i \to \infty$) and $c_i/\theta_i$ (when $\hat{\theta}_i = 0$).

We consider the following cases:

1. $\hat{\alpha}_i < c_i/\theta_i \;\Rightarrow\; (\hat{c}_i)'_{\hat{\theta}_i} < 0 \;\Rightarrow\; \hat{c}_i$ is a decreasing function $\Rightarrow\; \inf_{\hat{\theta}_i} \hat{c}_i = \hat{\alpha}_i$, when $\hat{\theta}_i \to \infty$.
2. $\hat{\alpha}_i = c_i/\theta_i \;\Rightarrow\; (\hat{c}_i)'_{\hat{\theta}_i} = 0 \;\Rightarrow\; \hat{c}_i$ is a constant function $\Rightarrow\; \inf_{\hat{\theta}_i} \hat{c}_i = \hat{\alpha}_i = c_i/\theta_i,\ \forall \hat{\theta}_i \ge 0$.
3. $\hat{\alpha}_i > c_i/\theta_i \;\Rightarrow\; (\hat{c}_i)'_{\hat{\theta}_i} > 0 \;\Rightarrow\; \hat{c}_i$ is an increasing function $\Rightarrow\; \inf_{\hat{\theta}_i} \hat{c}_i = c_i/\theta_i$, when $\hat{\theta}_i = 0$,

as stated.

Proving $\hat{c}_i^* = \inf_{\hat{\theta}_i} \hat{c}_i$ is based on (4). The first line can be rewritten as

$$\max_{\bar{z},p,\hat{\theta}} \sum_{i=1}^{\mathcal{I}} \left[\hat{c}_i\left(\mu_i\bar{z}_i - \lambda_i(1-p_i)\right) - \lambda_i r_i p_i\right],$$

where $\mu_i\bar{z}_i - \lambda_i(1-p_i) \le 0$ by the problem definition and thus for any fixed $p_i$ we get that $\hat{c}_i = \inf_{\hat{\theta}_i} \hat{c}_i$ is optimal. Q.E.D.

***Proof of Proposition 1:*** First, we turn to the rejectable classes.

The proof follows from (4) by demonstrating that for any solution where $\bar{q}_i = a$, for some $a > 0$, a modified solution with $\bar{q}_i = 0$ and an increased $p_i$ (according to (2)) achieves a better value function.

For $\bar{q}_i = a > 0$ and $p_i = p^a$, we have

$$\lambda_i = \mu_i\bar{z}_i + \left(\theta_i+\hat{\theta}_i\right)a + \lambda_i p_i^a,$$

with the following value function:

$$F^a = \left(c_i+\hat{\alpha}_i\hat{\theta}_i\right)a + r_i\lambda_i p_i^a + \sum_{j\neq i}^{\mathcal{I}}\left[c_j\bar{q}_j + \hat{\alpha}_j\hat{\theta}_j\bar{q}_j + r_j\lambda_j p_j\right]$$

For $\bar{q}_i = 0$ and $p_i = p_i^0$, we have

$$\lambda_i = \mu_i\bar{z}_i + \lambda_i p_i^0,$$

with the following value function:

$$F^0 = r_i\lambda_i p_i^0 + \sum_{j\neq i}^{\mathcal{I}}\left[c_j\bar{q}_j + \hat{\alpha}_j\hat{\theta}_j\bar{q}_j + r_j\lambda_j p_j\right].$$

Equating expressions for $\lambda_i$, we get

$$p_i^0 = p_i^a + \frac{a}{\lambda_i}(\theta_i+\hat{\theta}_i).$$

Then, comparing the two value functions, we have

$$\begin{aligned}
F^a - F^0 &= \left(c_i + \hat{\alpha}_i\hat{\theta}_i\right) a + r_i\lambda_i p_i^a - r_i\lambda_i p_i^0 \\
&= \left(c_i + \hat{\alpha}_i\hat{\theta}_i\right) a + r_i\lambda_i p_i^a - r_i\lambda_i p_i^a - r_i a\left(\theta_i + \hat{\theta}_i\right) \\
&= \left(c_i + \hat{\alpha}_i\hat{\theta}_i\right) a - r_i a\left(\theta_i + \hat{\theta}_i\right) \\
&= a\left(\left(c_i + \hat{\alpha}_i\hat{\theta}_i\right) - r_i\left(\theta_i + \hat{\theta}_i\right)\right) \\
&\geq a\left(\left(c_i + \hat{\alpha}_i\hat{\theta}_i\right) - \hat{c}_i\left(\theta_i + \hat{\theta}_i\right)\right) = 0,
\end{aligned}$$

where the first inequality is derived from proposition's condition that $r_i \leq \hat{c}_i$, and the second is from the definition of $\hat{c}_i$. This proves that when $r_i \leq \hat{c}_i$, $\bar{q}_i = 0$ is preferable.

Note that in this case we can plug in $\bar{q}_i = 0$ into (2) and get that $\bar{z}_i = \frac{\lambda_i}{\mu_i}(1 - p_i)$, which is equivalent to $p_i = 1 - \frac{\mu_i}{\lambda_i}\bar{z}_i$.

Lastly, we move to the non-rejectable classes. Since $\hat{c}_i^* < r_i$ then $p_i^* = 0$ follows from (4). Q.E.D.

***Proof of Theorem 1:*** Let us separately consider the rejectable classes $L \subset \mathcal{I}$ and the non-rejectable classes $K \subset \mathcal{I}$. Note that $L \cup K = \mathcal{I}$ and $L \cap K = \varnothing$. For each rejectable class $l \in L$ (i.e., $\mathcal{L}_l = r_l$), we substitute $p_l = 1 - \frac{\mu_l \bar{z}_l}{\lambda_l}$ (as derived in Proposition 1) into the objective function of (4). For each non-rejectable class $k \in K$ (i.e., $\mathcal{L}_k = \hat{\alpha}_k$ or $\mathcal{L}_k = \frac{c_k}{\theta_k}$), we substitute $p_k = 0$ (as derived in Proposition 1). We, therefore, have:

$$\begin{aligned}
&\min_{\bar{z},p,\hat{\theta}} \left[\sum_{l\in L}(\lambda_l r_l - r_l\mu_l\bar{z}_l) + \sum_{k\in K}\left(\lambda_k \inf_{\hat{\theta}_k}\hat{c}_k - \inf_{\hat{\theta}_k}\hat{c}_k\mu_k\bar{z}_k\right)\right] \Leftrightarrow \quad \max_{\bar{z}}\left[\sum_{l\in L}\mathcal{L}_l\mu_l\bar{z}_l + \sum_{k\in K}\mathcal{L}_k\mu_k\bar{z}_k\right] \\
&\text{s.t.} \quad \sum_{i\in\mathcal{I}}\bar{z}_i \leq 1, \quad \lambda_i/\mu_i \geq \bar{z}_i \geq 0,
\end{aligned}$$

where equivalence is due to constant elimination and $\mathcal{L}$ definition. Clearly, the optimal solution for the above problem is to allocate the servers according to their $\mathcal{L}\mu$ index. Q.E.D.

***Proof of Lemma 2:*** Let us consider the class that does not satisfy SL requirement given original $\mathcal{L}\mu$ index for the class. In order to satisfy the requirement we consider the trade-off between reallocating capacity from other classes and using our controls to reduce the amount of clients waiting in the queue. Satisfying the constraint for such class means that $\hat{z} = \lambda(1-p)\left(1 - (\theta + \hat{\theta})\tau\right)/\mu$. Let us take derivatives:

$$\hat{z}'_p = -\frac{\lambda}{\mu}\left(1 - (\theta + \hat{\theta})\tau\right); \quad \hat{z}'_{\hat{\theta}} = -\frac{\lambda}{\mu}(1-p)\tau,$$

where $\hat{z}'_p \leq 0$ because $\hat{z} \geq 0$ meaning that $1 - (\theta + \hat{\theta})\tau \geq 0$.

Both derivatives show that the effectiveness of one control is getting smaller when the other control is used more, meaning that if one of the controls is more effective (cost-wise) than the other when both are zeros, it will always be more effective. Q.E.D.

***Proof of Theorem 2:*** The idea of the proof is similar to that of Theorem 1. However, now we must consider the CB classes separately. Let $L \subset \mathcal{I}$ denote all rejectable classes (i.e., such $l \in \mathcal{I}$ that $\mathcal{L}_l = r_l$) and $K^{SL} \subset \mathcal{I}$ denote those non-rejectable classes $k \in \mathcal{I}$ that have either $\mathcal{L}_k = c_k/\theta_k$ with $\tau_k \geq 1/\theta_k$ or $\mathcal{L}_k = \hat{\alpha}_k$.

Additionally, we consider the CB classes $\mathcal{I}^{CB} \subset \mathcal{I}$ (i.e., such that for all $m \in \mathcal{I}^{CB}$, $\mathcal{L}_m = c_m/\theta_m$ and $\tau_m < 1/\theta_m$). Note that

$$L \cup K^{SL} \cup \mathcal{I}^{CB} = \mathcal{I} \quad \text{and} \quad L \cap K^{SL} = K^{SL} \cap \mathcal{I}^{CB} = \mathcal{I}^{CB} \cap L = \varnothing.$$

For each class $l \in L$ we insert $p_l = 1 - \mu_l \bar{z}_l/\lambda_l$ into the objective function of (8). For each class $k \in K^{SL}$ we insert $p_k = 0$:

$$\begin{aligned}
&\max_{\bar{z},p,\hat{\theta}} \left[ \sum_{l\in L} (r_l \mu_l \bar{z}_l - \lambda_l r_l) + \sum_{k\in K^{SL}} (\mathcal{L}_k \mu_k \bar{z}_k - \lambda_k \mathcal{L}_k) + \sum_{m\in\mathcal{I}^{CB}} (\hat{c}_m \mu_m \bar{z}_m + \lambda_m (\hat{c}_m - r_m) p_m - \lambda_m \hat{c}_m) \right] \\
\Leftrightarrow \quad &\max_{\bar{z},p} \left[ \sum_{l\in L} r_l \mu_l \bar{z}_l + \sum_{k\in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \sum_{m\in\mathcal{I}^{CB}} (\hat{c}_m \mu_m \bar{z}_m - \lambda_m (r_m - \hat{c}_m) p_m - \lambda_m \hat{c}_m) \right] \\
&\text{s.t.} \quad \sum_{i\in\mathcal{I}} \bar{z}_i \le 1, \quad \frac{\lambda_i}{\mu_i} \ge \bar{z}_i \ge 0, \ i \in \mathcal{I} \\
&\qquad \bar{z}_m > \lambda_m (1-p_m)(1-(\theta_m + \hat{\theta}_m)\tau_m)/\mu_m \text{ for } m \in \mathcal{I}^{CB};
\end{aligned}$$

Let us look closer at the CB classes. According to Lemma 2 in optimal solution for each CB class $m_0$ we have two cases: use only rejection control ($\hat{\theta}_{m_0} = 0$) and use only controlled abandonment ($p_{m_0} = 0$). We denote $M := \sum_{m\in\mathcal{I}^{CB}|m\neq m_0} (\hat{c}_m \mu_m \bar{z}_m - \lambda_m (r_m - \hat{c}_m) p_m - \lambda_m \hat{c}_m)$

*Rejection case*: $\hat{\theta}_{m_0} = 0$. We note that for any fixed value of $\bar{z}_{m_0}$, minimizing $p_{m_0}$, results in two cases:

**(a)** $\boldsymbol{\bar{z}_{m_0} < (1 - \theta_{m_0}\tau_{m_0})\lambda_{m_0}/\mu_{m_0}}$: The optimal solution is

$$p_{m_0} = 1 - \frac{\mu_{m_0} \bar{z}_{m_0}}{\lambda_{m_0}(1 - \theta_{m_0}\tau_{m_0})}$$

Substituting this into our objective function will yield the following:

$$\max_{\bar{z}} \left[ \sum_{l\in L} r_l \mu_l \bar{z}_l + \sum_{k\in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \left( \frac{c_{m_0}}{\theta_{m_0}} \mu_{m_0} \bar{z}_{m_0} - \lambda_{m_0} r_{m_0} + \frac{r_{m_0} - \frac{c_{m_0}}{\theta_{m_0}}}{1 - \theta_{m_0}\tau_{m_0}} \mu_{m_0} \bar{z}_{m_0} \right) + M \right],$$

where the middle addend for the CB classes is a constant and does not affect the solution. We, therefore, get that

$$\max_{\bar{z}} \left[ \sum_{l\in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k\in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \left( \mathcal{L}_{m_0} + \frac{r_{m_0} - \mathcal{L}_{m_0}}{1 - \theta_{m_0}\tau_{m_0}} \right) \mu_{m_0} \bar{z}_{m_0} + M \right].$$

**(b)** $\boldsymbol{\bar{z}_{m_0} \ge (1 - \theta_{m_0}\tau_{m_0})\lambda_{m_0}/\mu_{m_0}}$: The optimal solution would be $p_{m_0} = 0$, so the objective function will be:

$$\max_{\bar{z}} \left[ \sum_{l\in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k\in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \mathcal{L}_{m_0} \mu_{m_0} \bar{z}_{m_0} + M \right],$$

*Controlled abandonment case*: $p_{m_0} = 0$. We note that for any fixed value of $\bar{z}_{m_0}$, we want to minimize $\hat{c}_{m_0}$ due to $\bar{z}_{m_0} \le \lambda_{m_0}/\mu_{m_0}$, meaning minimizing $\hat{\theta}_{m_0}$. This again results in two similar cases:

**(a)** $\boldsymbol{\bar{z}_{m_0} < (1 - \theta_{m_0}\tau_{m_0})\lambda_{m_0}/\mu_{m_0}}$: The optimal solution is

$$\hat{\theta}_{m_0} = \frac{1}{\tau_{m_0}} - \frac{\mu_{m_0} \bar{z}_{m_0}}{\lambda_{m_0}\tau_{m_0}} - \theta_{m_0}$$

Substituting this into our objective function will yield the following:

$$\max_{\bar{z}} \left[ \sum_{l\in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k\in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \left( \frac{c_{m_0} + \frac{\hat{\alpha}_{m_0}}{\tau_{m_0}} \left(1 - \frac{\mu_{m_0}}{\lambda_{m_0}} \bar{z}_{m_0}\right) - \hat{\alpha}_{m_0} \theta_{m_0}}{\frac{1}{\tau_{m_0}} \left(1 - \frac{\mu_{m_0}}{\lambda_{m_0}} \bar{z}_{m_0}\right)} \left(1 - \frac{\mu_{m_0}}{\lambda_{m_0}} \bar{z}_{m_0}\right) (-\lambda_{m_0}) \right) + M \right],$$

from where we get

$$\max_{\bar{z}} \left[ \sum_{l \in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k \in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \left( c_{m_0} \tau_{m_0} + \hat{\alpha}_{m_0} \left( 1 - \frac{\mu_{m_0}}{\lambda_{m_0}} \bar{z}_{m_0} \right) - \hat{\alpha}_{m_0} \theta_{m_0} \tau_{m_0} \right) (-\lambda_{m_0}) + M \right],$$

and then, by eliminating constants, to an equivalent problem

$$\max_{\bar{z}} \left[ \sum_{l \in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k \in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \hat{\alpha}_{m_0} \mu_{m_0} \bar{z}_{m_0} + M \right].$$

**(b)** $\boldsymbol{\bar{z}_{m_0} \geq (1 - \theta_{m_0} \tau_{m_0}) \lambda_{m_0} / \mu_{m_0}}$: The optimal solution would be $\hat{\theta}_{m_0} = 0$, so the objective function will be:

$$\max_{\bar{z}} \left[ \sum_{l \in L} \mathcal{L}_l \mu_l \bar{z}_l + \sum_{k \in K^{SL}} \mathcal{L}_k \mu_k \bar{z}_k + \mathcal{L}_{m_0} \mu_{m_0} \bar{z}_{m_0} + M \right],$$

Those cases mean that if the allocated capacity for a CB class $m$ is not enough to satisfy the SL constraint, we will choose the modified $\mathcal{L}^{\mathrm{S}}_m \mu_m$ index for that class, where $\mathcal{L}^{\mathrm{S}}_m$ is the minimum between $\hat{\alpha}_m = \mathcal{L}_m + (\hat{\alpha}_m - \mathcal{L}_m)$, corresponding to controlled abandonment, and $\mathcal{L}_m + \frac{r_m - \mathcal{L}_m}{1 - \tau_m \theta_m}$, corresponding to rejection. Q.E.D.