OPERATIONS RESEARCH
Vol. 53, No. 6, November–December 2005, pp. 968–981
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5306 | 0968

informs

DOI 10.1287/opre.1050.0230

Managing Response Time in a Call-Routing Problem with Service Failure

Francis de Véricourt
Fuqua School of Business, Duke University, Durham, North Carolina 27708, fdv1@duke.edu

Yong-Pin Zhou
University of Washington Business School, Seattle, Washington 98195, yongpin@u.washington.edu

Traditional research on routing in queueing systems usually ignores service quality related factors. In this paper, we analyze the routing problem in a system where customers call back when their problems are not completely resolved by the customer service representatives (CSRs). We introduce the concept of call resolution probability, and we argue that it constitutes a good proxy for call quality. For each call, both the call resolution probability (p) and the average service time ($1/\mu$) are CSR dependent. We use a Markov decision process formulation to obtain analytical results and insights about the optimal routing policy that minimizes the average total time of call resolution, including callbacks. In particular, we provide sufficient conditions under which it is optimal to route to the CSR with the highest call resolution rate ($p\mu$) among those available. We also develop efficient heuristics that can be easily implemented in practice.

Subject classifications: dynamic programming/optimal control; Markov: infinite state; probability: stochastic model applications; queues: Markovian; multichannel.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received April 2003; revisions received March 2004, October 2004, November 2004; accepted December 2004.

1. Introduction

Customer service oriented call centers are traditionally operated as cost centers. Service accessibility and customer waiting time are the dominant performance measures. As a result, capacity planning and call-routing software systems strive to minimize costs while achieving self-imposed service-level constraints, such as "average wait in queue less than 15 seconds." These traditional approaches do not consider, however, the quality of answers provided by the customer service representatives (CSRs). Low quality of service has a significant impact on the call center operations besides customer defection: As dissatisfied customers call back for more help for the same problem, the load on the system increases.

This operational impact of service failure is often ignored by call center capacity planning and call-routing management systems. Our paper is motivated by the problems at a major European telecommunication service provider that found that, on average, a customer needed to talk to more than three different CSRs to get his/her problem resolved. This company also observed noticeable differences among CSRs in their ability to resolve the customers' problems. In our paper, we integrate this service quality related information into call-routing decisions. The goal is to minimize the average total time of call resolution, defined as the total time spent by a customer in the system to resolve one issue, including all the callbacks.

A key feature of our approach is the way we model the quality of a CSR's answer. For customer service call centers, a high-quality answer provided by the CSR should resolve the customer's issue during that call. We operationalize this concept by defining call quality as the call resolution probability, the probability that the customer is satisfied and does not call back for the same problem. The call resolution probability is directly related to a customer's perception of call quality, which depends on the CSR's understanding of the customer's needs, courtesy, and competency (Zeithaml et al. 1993). Furthermore, it can be quantified and measured by most of the call center information systems in use today.

Our experience suggests that a CSR's call resolution probability is often highly correlated with his/her call speed (defined as the service rate). On the one hand, the correlation could be negative. Due to very high turnover rates and long training lead time in this industry (see Gans and Zhou 2002 for example), some call centers are pressed to make the most use of their CSRs. It is common for the call center to compensate CSRs on the number of calls served over a period of time, or their call handle time, thereby encouraging them to handle calls as fast as possible. As a result, CSRs sometimes rush to end a call without making sure that the root problem is fixed and will not reoccur later (Read 2002). On the other hand, the correlation could be positive. Many times, better trained and more experienced

CSRs are able to handle the calls faster and provide higher service quality at the same time. In this paper, we model the service time and the call resolution probability as exogenous variables, and we do not explicitly model the correlation between them.

Intuitively, call centers that deal with complex issues, such as technical support for corporate computer users or medical help over the phone, may have low call resolution probabilities. Nevertheless, we know from our experience that even when customer problems are simple (as for the European call center on which this study is based), the call resolution probabilities can be significantly low. We believe that this comes from the CSR compensation system mentioned previously and the high turnover rate that results in undertrained employees. In this paper, we describe routing rules that account for these call resolution probabilities. Although we do not directly identify compensation schemes that can improve the call resolution probabilities while addressing the high turnover rate, our results provide interesting insights into this issue.

We analyze a call-routing problem where there exist several classes of CSRs, each with its own average call speed μ and call resolution probability p. The goal is to minimize the average total time of call resolution. Potentially, there is a trade-off between call speed and call resolution in routing calls: If call resolution is the only concern, then it would be optimal to route calls only to the CSR class with the highest p. The customers' wait, however, may become excessively long. If call speed is the only concern, then the objective would be to minimize the average waiting time of each call instance independently, without paying attention to the number of customer attempts. Hence, we feel the average total time of call resolution is the best single measure that encompasses both call speed and call resolution, and it can be construed as the average number of customer tries times the average waiting time of each try. Other objective functions, such as linear combinations of call resolution and call speed, are possible, but the weights are hard to determine and they generally lead to intractable models.

We formulate the routing problem as a Markov decision process (MDP), where the call center is represented by a heterogeneous, multiserver queueing system. In this framework, we provide several partial characterizations of the optimal routing policy. Our main result states that, whenever possible, a call should be routed to the CSR class with the highest call resolution rate, $p\mu$. If the highest-$p\mu$ CSRs are all busy, then the call may be routed to another available CSR or kept in the queue. Furthermore, we derive sufficient conditions under which it is optimal to route a call to the CSR with the highest resolution rate *among the available CSRs*. We call this the $p\mu$ rule. In particular, we show that when the CSRs differ only in their call speed or call resolution probability, the $p\mu$ rule is optimal. We also fully characterize the optimal routing policy for a system with two heterogeneous CSRs. In this case, we show that the optimal policy is of a threshold type: A call will always be routed to the CSR with the higher resolution rate whenever possible; the other CSR will be routed a call only when the number of calls waiting in queue exceeds a certain threshold.

Based on these findings, we propose simple and intuitive routing policies. Our numerical studies show that the $p\mu$ rule performs very well in most cases, even when it is not optimal. Moreover, the $p\mu$-t policy, defined as the $p\mu$ rule plus a threshold, is almost optimal in all of our test cases. We also numerically demonstrate that call centers can significantly improve their performance by incorporating call resolution probability p into routing decisions.

The $p\mu$ index introduced in this paper is a simple and effective routing index that accounts for both the call speed and the call quality. It also suggests that CSRs should be evaluated and compensated on their call resolution rate, rather than their service rate alone, as is often the case.

To ascertain the robustness of our findings, we analyze the problem under more general modeling assumptions. We show that our results remain valid when callbacks are put in a separate queue and given priority. We also show numerically that the $p\mu$-based policies perform well even if there is an exponentially distributed delay before a customer calls back. When the service time depends on whether, and how many times, the customer has talked to the same CSR before, we introduce and evaluate a dedicated routing policy, which routes new calls using the $p\mu$-t policy, but always routes callbacks to the same CSR. A requirement for the implementation of this policy is the call center's ability to identify the history of a call before serving it (e.g., a case number is required for callbacks at the phone prompt), which is not the case for the call center we study. In §5.4, we will study the dedicated policy as an extension to the basic model.

The rest of this paper is organized as follows: In §2, we review the literature, and in §3, we formulate and discuss the model. Results for the optimal routing policy are presented in §4. In §5, we use extensive numerical tests to show the importance of accounting for call resolution probability in making the routing decisions. Several heuristics are proposed and compared. We also analyze the problem when some modeling assumptions are relaxed. We conclude the paper and comment on further research in §6.

2. Literature Review

The probability of health deterioration after treatment in the health care system (e.g., Berk and Moinzadeh 1998, De Angelis 1998), which is a strong indicator of the treatment efficiency, is similar to the probability of callback, $1-p$, in our model. To our knowledge, however, our paper is the first to apply such a measure of quality to the research of call centers or other service delivery systems.

If calls bring direct revenue to the company (e.g., catalog merchant), customer loyalty, measured by the probability of

defection, better reflects the service quality provided by the call center. Hall and Porteus (2000) and Gans (2002) are two examples of this approach. In this paper, we focus on the customer service call centers, so we assume that dissatisfied customers will call back, instead of simply defecting. Furthermore, to address customer allocation and capacity-planning problems, we use a more detailed model of the service system than those in Hall and Porteus (2000) and Gans (2002).

There is a large body of literature on the retrial queues. See Falin and Templeton (1997) and the references therein. More recently, Mandelbaum et al. (e.g., 1999a, b; 2000) study the effect of customer retrial behavior patterns specifically in the context of call centers. The customer retrials they study differ from the customer callbacks in this paper in that a retrial occurs *before* the customer receives her service (when a customer calls and receives a busy signal, she "retries" by calling back sometime later), while a callback occurs *after* the customer has already received her service.

When the CSRs in a call center have different skills and speeds, skills-based routing has been shown to outperform the first-come-first-served (FCFS) and first-available-CSR call-routing rules in many situations. As a result, much study has been done on the skills-based call-routing schemes, both in the industry and in academia (e.g., Bell and Williams 2001, Harrison and Lopéz 1999, Gans and Zhou 2003, Atar et al. 2004, and some other references contained in Gans et al. 2003).

Research on routing in general often suggests priority-based policies: Some call-CSR combinations are given priority so they will be used whenever possible; the other combinations will be used only if the system is in certain states. A good example is the traditional $c\mu$ rule (see Van Mieghem 1995 for a generalized $c\mu$ rule and Mandelbaum and Stolyar 2002 for its application in the call center setting). The main issue in these models is how to minimize total cost based on the different processing speeds associated with each call-CSR combination and the different call-type specific holding costs.

The stream of research most relevant to ours is the so-called slow-server problem. In the two-server slow-server problem, there is one Poisson arrival stream and two heterogeneous exponential servers. The objective is to find a routing policy to minimize the average wait. Larsen (1981) first formulates the problem and conjectures that a threshold policy should be optimal. Later, Lin and Kumar (1984), Walrand (1984), and Koole (1995) prove this conjecture using MDP policy iteration, coupling argument, and MDP value iteration, respectively. Larsen and Agrawala (1983) develop a good and computationally simple approximation to the threshold.

The general slow-server problem allows for more than two heterogeneous servers. Due to the increase in state-space dimensionality, the problem becomes very complex (e.g., see Rykov 2001, Luh and Viniotis 2002). So far, the optimal routing policy has not been fully characterized for the general case (see de Véricourt and Zhou 2005). Our model can be viewed as the general slow-server model with multiple classes of servers and the additional callback loops—in particular, when the call resolution probabilities are all equal to 1, our model reduces to the general slow-server problem. The optimality of the $p\mu$ rule in our model implies that allocating a call to the fastest server (the μ rule) is optimal for the general slow-server problem. This extends the existing literature on the general slow-server problem.

Most analysis of the slow-server problem is exact. Teh and Ward (2002), on the other hand, study the problem in the heavy-traffic regime. They show that, as the heavy-traffic limit is approached, the system is stable and the threshold policy is optimal if and only if the threshold grows at a logarithmic rate. In other words, in the heavy-traffic regime, the threshold does not disappear.

3. Formulation of the Problem

3.1. Model and Assumptions

Consider a call center with C classes of CSRs. A class is a group of CSRs with the same service time distribution and call resolution probability. We assume that there are S_i CSRs in class i, $i \in \{1, \ldots, C\}$. For a Class-i CSR, $i \in \{1, \ldots, C\}$, the service time is exponentially distributed with rate μ_i, and the call resolution probability is p_i. When a Class-i CSR completes a call, there are two possible outcomes: (1) With probability p_i, the issue is completely resolved and the customer will simply leave the system, and (2) with probability $1 - p_i$, the issue is not completely resolved, and the customer calls back right away.

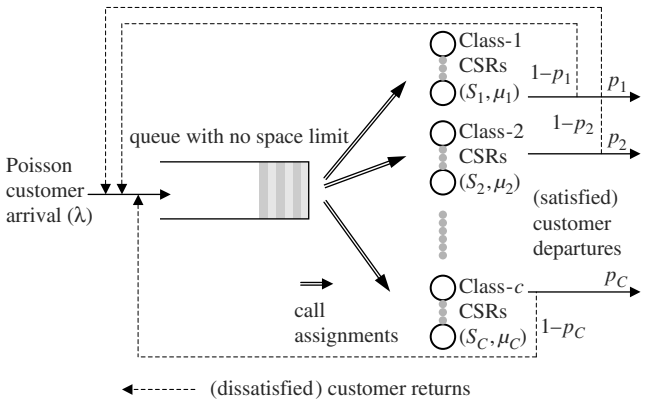
Our model does not differentiate new calls from callbacks, and all customers are served on an FCFS. In practice, however, callbacks are sometimes given higher priority if they can be identified. This means that callbacks are put in a separate queue and given priority over new calls. A simple coupling argument shows, nonetheless, that such a priority scheme does not alter the average total waiting time of the system, and our findings remain valid.

The arrival of customers with new requests follows a Poisson process with rate λ, and they wait in a queue if they are not served upon arrival. There is no limit on the waiting space. To ensure stability, we assume that $\lambda < \sum_{i=1}^{C} S_i p_i \mu_i$. See Figure 1 for details.

Due to the memoryless property of Poisson arrival and exponential service times, the state of the system at time t can be described by a $(C+1)$-dimensional vector $\mathbf{n}(t) = (n_0(t), n_1(t), \ldots, n_C(t))$, where $n_0(t) \geqslant 0$ is the number of calls waiting in the queue and $n_i(t) \in \{0, \ldots, S_i\}$, $i \in \{1, \ldots, C\}$ is the number of busy Class-i CSRs.

At any time, the system controller must decide (1) whether to keep a call in the queue or to route it to an available CSR, and (2) if a call is to be routed, to which CSR class it should be routed. The goal of our model is to minimize the average total time of call resolution.

Figure 1. Model overview.



In this model, we assume that both the call resolution probabilities and the service rates are independent of the number of previous calls made by the customer for the same problem. Such an assumption may not be realistic in certain situations, for instance, when there is a setup time each time a customer meets a new CSR, or more generally when the service time decreases with the number of attempts. We discuss this situation in §5.3.

We also assume that customers *immediately* return to the system when they are dissatisfied. This assumption is reasonable when a customer can quickly check the accuracy of the CSR's answer. Examples include technical support call centers that deal with computer hardware/software applications, where the delay in callback is usually small compared to the service time. In other practical situations, however, dissatisfied customers call back after a longer delay. In §5.4, we present a model where an exponential amount of time elapses before dissatisfied customers call back. Numerical studies show that the routing policies developed for the immediate callback model also perform well in this case.

3.2. The Markov Decision Problem

The routing policies we study are nonanticipating and nonpreemptive. Furthermore, due to the Markovian assumptions, the policies are also not history dependent. As is well known in the literature, it is optimal to take actions only at arrival and service departure epochs. Any possible action is represented by a C-dimensional vector $(a_1, \ldots, a_C)$, where a_i, $\forall i \in \{1, \ldots, C\}$, is the number of calls routed to Class-i CSRs. In particular, the zero vector represents the (non-)action of not routing any call. A routing policy π is thus a rule that determines, for every decision epoch, what action to take.

The objective is to determine the routing policy that minimizes the average total time of call resolution. By Little's Law, this is equivalent to minimizing the average number of customers in the system. As a result, we look for the Markov routing policies that minimize the average number of customers in the system:

$$g^* = \min_{\pi} \limsup_{T\to\infty} \frac{1}{T} E^{\pi}_{\mathbf{n}_0}\left[\int_0^T \sum_{i=0}^{C} \mathbf{n}_i(t)\, dt\right], \tag{1}$$

where $E^{\pi}_{\mathbf{n}_0}$ denotes the conditional expectation given policy π and the initial state at time 0.

The main approach we use in this paper is the standard MDP value iteration (e.g., see Ha 1997 or Veatch and Wein 1996). Let $v(\mathbf{n})$ be the standard MDP "cost-to-go" function in state $\mathbf{n}$, then v is a mapping from $\mathbf{N}^{C+1}$ to $\Re^+$, where $\mathbf{N}$ and $\Re^+$ are the sets of integers and nonnegative real numbers, respectively. In the next section, we will define the desirable properties for the optimal MDP value function $v(\cdot)$ and show that these properties are preserved by the value iteration operators. We define below the two value iteration operators T and Γ.

Because the interarrival and service time are exponentially distributed, we can study an equivalent Markov process with independent, identically distributed (i.i.d.) interevent time by adding fictitious transitions. This procedure is known as *uniformization*. (See §11.5 in Puterman 1994 for details.) The uniformized Markov process will have a fixed total transition rate of $\lambda + \sum_{i=1}^{C} S_i\mu_i$ in every state. Without loss of generality, we can scale the time and assume that $\lambda + \sum_{i=1}^{C} S_i\mu_i = 1$. Let $\mathbf{e}_i$, $i \in \{0, \ldots, C\}$, denote a $(C+1)$-dimension vector whose $(i+1)$th component is 1 and all other components 0, $\Omega = \{f \mid f\colon \mathbf{N}^{C+1} \to \Re^+\}$, and $T\colon \Omega \to \Omega$.

We denote by $K(\mathbf{n})$ the set of classes with available CSRs in state $\mathbf{n}$:

$$K(\mathbf{n}) = \{i \in \{1, \ldots, C\} \mid n_i < S_i\}. \tag{2}$$

Then, for any $v \in \Omega$,

$$Tv(\mathbf{n}) = \begin{cases} v(\mathbf{n}) & \text{if } n_0 = 0 \text{ or } K(\mathbf{n}) = \varnothing, \\ \min\{Tv(\mathbf{n} + \mathbf{e}_j - \mathbf{e}_0), v(\mathbf{n}) \mid j \in K(\mathbf{n})\} & \\ \qquad \text{if } n_0 > 0 \text{ and } K(\mathbf{n}) \neq \varnothing. & \end{cases} \tag{3}$$

Note that more than one call can be routed at once. Instead of listing all these possible routings in the minimization operator, we choose to use an equivalent recursive definition in (3). The recursion is well defined because n_0 decreases by one each time a call is routed. For example, take $n_0 = 2$, and apply the previous recursive definition twice. We have

$$Tv(\mathbf{n}) = \min\big\{v(\mathbf{n} + \mathbf{e}_j + \mathbf{e}_k - 2\mathbf{e}_0), v(\mathbf{n} + \mathbf{e}_j - \mathbf{e}_0), \\ v(\mathbf{n}) \mid j \in K(\mathbf{n}),\ k \in K(\mathbf{n} + \mathbf{e}_j - \mathbf{e}_0)\big\}.$$

When $Tv(\mathbf{n}) = v(\mathbf{n} + \mathbf{e}_j + \mathbf{e}_k - 2\mathbf{e}_0)$ for some j, k, the corresponding policy routes two calls at the same time.

The MDP optimality equation then becomes

$$\Gamma v^*(\mathbf{n}) = v^*(\mathbf{n}) + g^*, \tag{4}$$

where g^* is the optimal average number of customers defined in (1), which is independent of the initial state (see, for instance, Rykov 2001 and the references therein), $v^*(\mathbf{n})$ is the optimal relative value function, and $\Gamma\colon \Omega \to \Omega$ is the dynamic operator that satisfies

$$\begin{aligned}\Gamma v(\mathbf{n}) = {} & \sum_{i=0}^{C} n_i + \lambda T v(\mathbf{n}+\mathbf{e}_0) \\ & + \sum_{i=1}^{C} n_i(1-p_i)\mu_i T v(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_0) \\ & + \sum_{i=1}^{C} n_i p_i \mu_i T v(\mathbf{n}-\mathbf{e}_i) + \sum_{i=1}^{C}(S_i-n_i)\mu_i T v(\mathbf{n}). \end{aligned} \tag{5}$$

Note that the last term corresponds to a fictitious transition due to the uniformization procedure. We allow an action at these transitions, as in Koole (1995).

4. Analysis of the Optimal Routing Policy

In this section, we present partial characterizations of the optimal routing policy. We first show that, whenever possible, it is optimal to route a call to the CSR class with the highest call resolution rate $p\mu$. We then derive conditions under which a generalization of this property, the $p\mu$ rule, is optimal. More precisely, we assume without loss of generality that the different classes of CSRs are indexed such that $p_1\mu_1 \geqslant \cdots \geqslant p_C\mu_C$. Then, the $p\mu$ rule stipulates that, if the state is $\mathbf{n}$ when a call is routed, then the call should be routed to CSR class $m(\mathbf{n})$, where $m(\mathbf{n}) = \min\{k \mid k \in K(\mathbf{n})\}$.

We end this section with a full characterization of the optimal routing policy for the two-CSR case.

4.1. Partial Characterization of the Optimal Policy

For any $v \in \Omega$, define

$$\Delta_i v(\mathbf{n}) = v(\mathbf{n}+\mathbf{e}_i) - v(\mathbf{n}) \quad \forall\, i \in \{0,\ldots,C\},$$
$$\Delta_{ij} v(\mathbf{n}) = v(\mathbf{n}+\mathbf{e}_i) - v(\mathbf{n}+\mathbf{e}_j) \quad \forall\, i,j \in \{0,\ldots,C\}.$$

Moreover, define V to be the set of all $v \in \Omega$ that satisfy the following properties:

Property 1. $\Delta_i v(\mathbf{n}) \geqslant 0\ \forall\, i \in K(\mathbf{n})$.

Property 2. $\Delta_0 v(\mathbf{n}) \geqslant 0$.

Property 3. $\Delta_{1i} v(\mathbf{n}) \leqslant 0$ if $1 \in K(\mathbf{n})$ and $i \in K(\mathbf{n})$.

Property 4. $\Delta_{10} v(\mathbf{n}) \leqslant 0$ if $1 \in K(\mathbf{n})$.

Properties 1 and 2 are fairly intuitive. They state that fewer calls in the system, either with Class-i CSRs or in queue, always result in smaller average total time in the system. Properties 3 and 4 together imply that, whenever possible, the policy corresponding to $v \in V$ always routes a call to a Class-1 CSR first.

The following lemma is used repeatedly in our analysis later. Its proof is straightforward, and thus omitted.

Lemma 1. *Let $\{x_1,\ldots,x_p\}$ and $\{y_1,\ldots,y_q\}$ be two sets of real numbers. If for any $i \in \{1,\ldots,p\}$ there exists a $j(i) \in \{1,\ldots,q\}$ such that $x_i \geqslant y_{j(i)}$, then $\min_{i\in\{1,\ldots,p\}}\{x_i\} \geqslant \min_{j\in\{1,\ldots,q\}}\{y_j\}$.*

The following lemma states that operator T preserves V.

Lemma 2. *If $v \in V$, then $Tv \in V$.*

Proof. In this proof, the terms "positive" and "negative" mean "nonnegative" and "nonpositive," respectively. Let $v \in V$. We first show that Tv satisfies Properties 1–3 by induction on n_0, the number of calls waiting in queue. We then deduce Property 4.

Step 1. Consider states $\mathbf{n}$ where $n_0 = 0$. A direct computation leads to $\Delta_i Tv(\mathbf{n}) = \Delta_i v(\mathbf{n})$, $\Delta_0 Tv(\mathbf{n}) = \min\{\Delta_i v(\mathbf{n}), \Delta_0 v(\mathbf{n}) \mid i \in K(\mathbf{n})\}$, and $\Delta_{1i} Tv(\mathbf{n}) = \Delta_{1i} v(\mathbf{n})$. It follows from $v \in V$ that Tv satisfies Properties 1–3 for $\mathbf{n}$ such that $n_0 = 0$.

Step 2. Consider states $\mathbf{n}$ where $n_0 > 0$. Assume that Tv satisfies Properties 1–3 for all states where the number of calls waiting in queue is strictly less than n_0.

Property 1. By definition, $Tv(\mathbf{n}+\mathbf{e}_i) = \min\{Tv(\mathbf{n}+\mathbf{e}_i+\mathbf{e}_j-\mathbf{e}_0), v(\mathbf{n}+\mathbf{e}_i) \mid j \in K(\mathbf{n}+\mathbf{e}_i)\}$. Note that if $j \in K(\mathbf{n}+\mathbf{e}_i)$, then $j \in K(\mathbf{n})$. Moreover, $Tv(\mathbf{n}+\mathbf{e}_i+\mathbf{e}_j-\mathbf{e}_0) \geqslant Tv(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0)$, because Tv is assumed to satisfy Property 1 for states with n_0-1 waiting calls. Furthermore, $v(\mathbf{n}+\mathbf{e}_i) \geqslant v(\mathbf{n})$ because $v \in V$. Hence, $Tv(\mathbf{n})$ satisfies Property 1 by Lemma 1.

Property 2. Similarly, because $K(\mathbf{n}+\mathbf{e}_0) = K(\mathbf{n})$, v satisfies Property 2, and Tv satisfies Property 2 for n_0-1, we can use Lemma 1 to show that $Tv(\mathbf{n}+\mathbf{e}_0) = \min\{Tv(\mathbf{n}+\mathbf{e}_j), v(\mathbf{n}+\mathbf{e}_0) \mid j \in K(\mathbf{n}+\mathbf{e}_0)\} \geqslant \min\{Tv(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0), v(\mathbf{n}) \mid j \in K(\mathbf{n})\} = Tv(\mathbf{n})$.

Property 3. For $j \in K(\mathbf{n}+\mathbf{e}_i)$, $j \neq 1$, we also have $j \in K(\mathbf{n}+\mathbf{e}_1)$. Therefore, $Tv(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0+\mathbf{e}_i) \geqslant Tv(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0+\mathbf{e}_1)$ because Tv is assumed to satisfy Property 3 for n_0-1. For $j = 1 \in K(\mathbf{n}+\mathbf{e}_i)$, we can choose $i \in K(\mathbf{n}+\mathbf{e}_1)$, and we have $Tv(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_0+\mathbf{e}_i) = Tv(\mathbf{n}+\mathbf{e}_i-\mathbf{e}_0+\mathbf{e}_1)$. Moreover, $v(\mathbf{n}+\mathbf{e}_i) \geqslant v(\mathbf{n}+\mathbf{e}_1)$, because v satisfies Property 3. Therefore, by Lemma 1, $Tv(\mathbf{n})$ satisfies Property 3. It follows that Tv satisfies Properties 1–3 for all n_0.

Property 4. Finally, note that

$$\begin{aligned} Tv(\mathbf{n}+\mathbf{e}_0) &= \min\{Tv(\mathbf{n}+\mathbf{e}_j), v(\mathbf{n}+\mathbf{e}_0) \mid j \in K(\mathbf{n}+\mathbf{e}_0)\} \\ &= \min\{Tv(\mathbf{n}+\mathbf{e}_1), v(\mathbf{n}+\mathbf{e}_0)\} = Tv(\mathbf{n}+\mathbf{e}_1). \end{aligned} \tag{6}$$

Therefore, $\Delta_{10} Tv(\mathbf{n}) = 0$. The second equality in (6) holds because Tv satisfies Property 3. The last one follows from

the fact that $Tv(\mathbf{n}+\mathbf{e}_1)$ is less than or equal to $v(\mathbf{n}+\mathbf{e}_1)$ from the definition of T, which is in turn less than or equal to $v(\mathbf{n}+\mathbf{e}_0)$ from Property 4. □

If $v \in V$, then according to Lemma 2, Tv satisfies Properties 3 and 4, and (3) becomes: For any $\mathbf{n}$ where $1 \in K(\mathbf{n})$,

$$Tv(\mathbf{n}) = \begin{cases} v(\mathbf{n}) & \text{if } n_0 = 0, \\ Tv(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_0) & \text{if } n_0 > 0. \end{cases} \tag{7}$$

In (6), we have actually shown that Tv satisfies a property stronger than Property 4.

COROLLARY 1. *If* $v \in V$, *then* $\Delta_{10}Tv(\mathbf{n}) = 0 \ \forall \mathbf{n}$ *s.t.* $1 \in K(\mathbf{n})$.

The following theorem establishes Properties 1–4 for the optimal value function.

THEOREM 1. *If* $p_1\mu_1 \geqslant p_i\mu_i \ \forall i \in \{2,\dots,C\}$ *and* $v \in V$, *then* $\Gamma v \in V$.

PROOF. Consider $v \in V$. We first study the sign of $\Delta_i\Gamma$ for $i \geqslant 1$. From (5),

$$\begin{aligned}\Delta_i\Gamma v(\mathbf{n}) = 1 &+ \left[(S_i - n_i - 1)\mu_1 + \sum_{j\neq i}(S_j - n_i)\mu_j\right]\Delta_i Tv(\mathbf{n}) \\ &+ \lambda\Delta_i Tv(\mathbf{n}+\mathbf{e}_0) + \mu_i(1-p_i)\Delta_0 Tv(\mathbf{n}) \\ &+ \sum_{j=1}^{C} n_j p_j \mu_j \Delta_i Tv(\mathbf{n}-\mathbf{e}_j) \\ &+ \sum_{j=1}^{C} n_j(1-p_j)\mu_j\Delta_i Tv(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_0), \end{aligned} \tag{8}$$

which is positive from Properties 1 and 2 and Lemma 2.

Similarly, based on Property 3 and Lemma 2, we conclude

$$\begin{aligned}\Delta_0\Gamma v(\mathbf{n}) = 1 &+ \sum_{j=1}^{C}(S_j - n_j)\mu_j\Delta_0 Tv(\mathbf{n}) + \lambda\Delta_0 Tv(\mathbf{n}+\mathbf{e}_0) \\ &+ \sum_{j=1}^{C} n_j p_j \mu_j \Delta_0 Tv(\mathbf{n}-\mathbf{e}_j) \\ &+ \sum_{j=1}^{C} n_j(1-p_j)\mu_j\Delta_0 Tv(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_0) \geqslant 0. \end{aligned} \tag{9}$$

We now turn our attention to $\Delta_{1i}\Gamma$:

$$\begin{aligned}\Delta_{1i}\Gamma v(\mathbf{n}) = &\left[(S_i - n_i - 1)\mu_i + \sum_{j\neq i}(S_j - n_j)\mu_j\right]\Delta_{1i} Tv(\mathbf{n}) \\ &+ (\mu_1 - \mu_i)[Tv(\mathbf{n}+\mathbf{e}_0) - Tv(\mathbf{n}+\mathbf{e}_1)] \\ &- (p_1\mu_1 - p_i\mu_i)\Delta_0 Tv(\mathbf{n}) + \lambda\Delta_{1i} Tv(\mathbf{n}+\mathbf{e}_0) \\ &+ \sum_{j=1}^{C} n_j p_j \mu_j \Delta_{1i} Tv(\mathbf{n}-\mathbf{e}_j) \\ &+ \sum_{j=1}^{C} n_j(1-p_i)\mu_j\Delta_{1i} Tv(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_0). \end{aligned} \tag{10}$$

We know that $Tv(\mathbf{n}+\mathbf{e}_0) = Tv(\mathbf{n}+\mathbf{e}_1)$ from Corollary 1. Moreover, because $p_1\mu_1 \geqslant p_i\mu_i$, $-(p_1\mu_1 - p_i\mu_i)\Delta_0 Tv(\mathbf{n}) \leqslant 0$ from Property 2. Consequently, by Property 3 and Lemma 2, $\Delta_{1i}\Gamma v(\mathbf{n}) \leqslant 0$. Finally, we compute $\Delta_{10}\Gamma v$:

$$\begin{aligned}\Delta_{10}\Gamma v(\mathbf{n}) = &\left[(S_1 - n_1 - 1)\mu_1 + \sum_{j>1}(S_j - n_j)\mu_j\right]\mu_j\Delta_{10} Tv(\mathbf{n}) \\ &+ \lambda\Delta_{10} Tv(\mathbf{n}+\mathbf{e}_0) - p_1\mu_1\Delta_0 Tv(\mathbf{n}) \\ &+ \sum_{j=1}^{C} n_j p_j \mu_j \Delta_{10} Tv(\mathbf{n}-\mathbf{e}_j) \\ &+ \sum_{j=1}^{C} n_j(1-p_j)\mu_j\Delta_{10} Tv(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_0), \end{aligned} \tag{11}$$

which is negative from Properties 2 and 4 and Lemma 2. □

Theorem 1 allows us to partially characterize the optimal policy:

COROLLARY 2. *Assume that* $p_1\mu_1 \geqslant p_i\mu_i \ \forall i \in \{2,\dots,C\}$. *It is optimal to route a call to a Class-*1 *CSR whenever possible.*

PROOF. From Theorem 1 and the application of MDP value iteration, the optimal value function can be shown to belong to V. Then, from Properties 3 and 4, we conclude that at any time, routing a call to a Class-1 CSR is better than either routing it to another available CSR or keeping it in the queue. □

It is worth noting that as long as $p_1\mu_1 \geqslant p_i\mu_i \ \forall i$, calls will be routed to a Class-1 CSR whenever possible, even when μ_1 is smaller than some μ_i. Hence, $p\mu$ is a more useful index than μ in routing decisions. We believe that managers should focus on improving the CSRs' call resolution rates $p\mu$, instead of just their service rates μ. Moreover, CSRs should be given incentives that correspond to their call resolution rate. For instance, CSRs' compensation could be evaluated based on "calls resolved" rather than "calls handled." Shumsky and Pinker (2003) have additional discussions on this topic.

4.2. The $p\mu$ Rule

Corollary 2 states that priority should be given to Class-1 CSRs, but it does not specify what to do when all Class-1 CSRs are busy and some other CSRs are available. A straightforward extension would be to give priority to the class with the highest $p\mu$ index among all those available. Recall that we name this the $p\mu$ rule.

In the case of two classes, the $p\mu$ rule is optimal, and can be viewed as an analog of the well-known $c\mu$ result. However, for more than two classes of CSRs, the $p\mu$ rule may not be optimal. Consider the case where Class-2 CSRs have a higher call resolution rate, but they are much slower (i.e., $p_2\mu_2 > p_3\mu_3$ and $\mu_2 \ll \mu_3$). In this case, a call routed

to a Class-2 CSR may still be in service when a better CSR (from Class 1) becomes available. However, if it had been routed to a Class-3 CSR instead, it might have either left the system earlier or have returned and been rerouted to a Class-1 CSR earlier. Therefore, the optimal policy may prefer Class 3 to Class 2 in some states. Specifically, for $C=3$, $S_1=5$, $S_2=S_3=2$, $\lambda=7$, $\mu_1=4$, $p_1=0.6$, $\mu_2=3$, $p_2=0.4$, $\mu_3=9$, and $p_3=0.1$, the optimal action in state $(1,5,0,1)$ is 3. That is, when one call is in the queue, all Class-1 CSRs are busy, and all Class-2 CSRs and 1 Class-3 CSR are available, it is optimal to route the call to a Class-3 CSR instead of a Class-2 CSR.

These are very rare and extreme cases, however. As our numerical tests will show, the $p\mu$ rule is optimal in most practical situations. Nevertheless, we need additional assumptions to analytically show the optimality of the $p\mu$ rule. Let the classes again be indexed such that $p_1\mu_1 \geqslant \cdots \geqslant p_C\mu_C$. We show below that the $p\mu$ rule is optimal when $\mu_2 \geqslant \cdots \geqslant \mu_C$. These conditions cover the cases in which the CSRs differ only in p (e.g., they follow the same scripts, but have different problem-solving skills/training) or only in μ (e.g., the slow-server problem). They also cover the cases in which p and μ are positively correlated (e.g., more experienced CSRs handle calls faster and give better answers).

Let W be the set of all real-valued functions defined on $\mathbf{N}^{C+1}$ that satisfy Properties 1, 2, 4, and the following property:

PROPERTY 3′. $\Delta_{ki}w(\mathbf{n}) \leqslant 0$ if $i \in K(\mathbf{n})$ and $k = m(\mathbf{n})$.

Property 3 is a special case of Property 3′ for $m(\mathbf{n})=1$, so W is a subset of V. In particular, under Property 3′ (7) remains true, and the policy corresponding to a value function belonging to W routes a call to a Class-1 CSR whenever possible.

The following lemma is analogous to Lemma 2.

LEMMA 3. *If $\mu_2 \geqslant \cdots \geqslant \mu_C$ and $w \in W$, then $Tw \in W$.*

PROOF. Because $W \subset V$, Tw satisfies Properties 1, 2, and 4 from Lemma 2. Now we use induction on n_0 to show that Tw satisfies Property 3′.

Step 1. When $n_0=0$, $\Delta_{ki}Tw(\mathbf{n}) = \Delta_{ki}w(\mathbf{n})$ by definition and Tw satisfies Property 3′.

Step 2. When $n_0>0$, we assume that Tw satisfies Property 3′ with n_0-1 calls waiting in the queue. This implies that $Tw(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0+\mathbf{e}_i) \geqslant Tw(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_0+\mathbf{e}_k)$ for all $j \in K(\mathbf{n}+\mathbf{e}_i)$, $j \neq k$. For $j=k \in K(\mathbf{n}+\mathbf{e}_i)$, we can choose $i \in K(\mathbf{n}+\mathbf{e}_k)$, and we have $Tw(\mathbf{n}+\mathbf{e}_k-\mathbf{e}_0+\mathbf{e}_i) = Tw(\mathbf{n}+\mathbf{e}_i-\mathbf{e}_0+\mathbf{e}_k)$. Furthermore, $w(\mathbf{n}+\mathbf{e}_i) \geqslant w(\mathbf{n}+\mathbf{e}_k)$ because $w \in V$, and the result follows from Lemma 1. □

We are now ready to provide sufficient conditions under which the $p\mu$ rule is optimal.

THEOREM 2. *If $p_1\mu_1 \geqslant \cdots \geqslant p_C\mu_C$, $\mu_2 \geqslant \cdots \geqslant \mu_C$ and $w \in W$, then $\Gamma w \in W$.*

PROOF. Because w satisfies Property 3′, it also satisfies Property 3. Following the same approach as in Theorem 1, we can show that Γw satisfies Properties 1, 2, and 4.

For Property 3′, a direct computation leads to, for $k \leqslant i$,

$$\begin{aligned}\Delta_{ki}\Gamma w(\mathbf{n}) = {} & \left[(S_i-n_i-1)\mu_i + \sum_{j\neq i}(S_j-n_j)\mu_j\right]\Delta_{ki}Tw(\mathbf{n}) \\ & + (\mu_k-\mu_i)[Tw(\mathbf{n}+\mathbf{e}_0) - Tw(\mathbf{n}+\mathbf{e}_k)] \\ & - (p_k\mu_k - p_i\mu_i)\Delta_0 Tw(\mathbf{n}) + \lambda\Delta_{ki}Tw(\mathbf{n}+\mathbf{e}_0) \\ & + \sum_{j=1}^{C} n_j p_j \mu_j \Delta_{ki} Tw(\mathbf{n}-\mathbf{e}_j) \\ & + \sum_{j=1}^{C} n_j(1-p_j)\mu_j\Delta_{ki}Tw(\mathbf{n}-\mathbf{e}_j+\mathbf{e}_0). \qquad (12)\end{aligned}$$

By the definition of T, $Tw(\mathbf{n}+\mathbf{e}_0) \leqslant Tw(\mathbf{n}+\mathbf{e}_k)$. Moreover, because μ_k is assumed to be larger than μ_i, $(\mu_k-\mu_i)\cdot[Tw(\mathbf{n}+\mathbf{e}_0) - w(\mathbf{n}+\mathbf{e}_k)] \leqslant 0$. The other terms of $\Delta_{ki}\Gamma$ are also negative from Properties 2 and 3′. □

The following straightforward corollary presents this result for the optimal control policy.

COROLLARY 3. *If $p_1\mu_1 \geqslant \cdots \geqslant p_C\mu_C$ and $\mu_2 \geqslant \cdots \geqslant \mu_C$, then*

- *The optimal policy routes a call to a Class-1 CSR whenever possible.*
- *If it is optimal to route a call in state $\mathbf{n}$, then this call is always routed to a Class-$m(\mathbf{n})$ CSR. That is, the $p\mu$ rule is optimal.*

By including callback loops, Corollary 3 provides nontrivial generalizations of the slow-server problem. Specifically, if we let $p_i=1$, $i \in \{1,\ldots,C\}$, then Corollary 3 extends the optimality of the μ rule in Lin and Kumar (1984), Walrand (1984), and Koole (1995) to more than two classes.

4.3. Threshold Policies

Results in §4.1 and §4.2 partially characterize the optimal policy. In particular, Corollaries 2 and 3 specify *where* to route a call when it is optimal to do so. They do not specify *when* to route a call. In most cases, a threshold policy seems to provide an efficient and simple way to make this type of decision. Optimality of the threshold policy has been proved for the two-server slow-server problem ($C=2$, $S_1=S_2=1$, and $p_1=p_2=1$) (see Lin and Kumar 1984, Walrand 1984, Koole 1995). The theorem below extends their result to include the callback loops (p_1 and p_2 less than 1). Its proof can be found in the online appendix.

THEOREM 3. *Suppose that $C=2$, $S_1=S_2=1$, and $p_1\mu_1 \geqslant p_2\mu_2$. The optimal routing policy is characterized by a threshold t^* such that*:

- *If the Class-1 CSR is available, the policy routes a waiting call to the Class-1 CSR*;

• *If the Class-*1 *CSR is busy, the policy routes a waiting call to the available Class-*2 *CSR if and only if the queue length is larger than* t^*.

Theorem 3 is the first optimality result for threshold policy in a queueing system with callback loops. It suggests that threshold-based policies are indeed suitable heuristics for systems with the callback loops.

When there are more than one CSR per class, and/or more than two classes, the situation is much more complex. Our extensive numerical tests (heavy, medium, and light traffic and different combinations of arrival and service rates, and call-resolution probabilities) suggest that the optimal policy is always a state-dependent threshold policy: In each state, it is optimal to route a call to a certain idle CSR (not necessarily following the $p\mu$ rule) if and only if queue length exceeds a threshold. These thresholds depend on the number of busy CSRs in *each* class, so potentially there could be as many as $[\prod_{i=2}^{C}(1+S_i)]-1$ thresholds.

5. Numerical Analysis and Extensions

Because optimal state-dependent threshold policies are hard to compute and incorporate, in §5.1 we propose heuristics that perform well and are simple to apply in practice. More fundamentally, we evaluate the importance of incorporating p into the routing decisions in §5.2. We also investigate various extensions of our modeling assumptions: In §5.3, we explore situations where the service rate depends on the number of times a CSR has talked to the same customer before. In §5.4, we consider cases in which customers call back, not immediately, but after an exponentially distributed time. To conclude our numerical analysis, we propose a lower bound system in §5.5.

5.1. $p\mu$-Based Policies

Lemmas 2 and 3 show that $p\mu$ is a very important routing index. In this section, we study two policies based on the $p\mu$ rule:

• Theorem 3 shows the optimality of threshold policies in simple settings that include callbacks. This inspires us to use a threshold-based policy for more complex settings. Consider the *$p\mu$ policy with a fixed threshold t*, or simply, the *$p\mu$-t policy*. With two CSR classes, this policy uses the $p\mu$ rule and routes a call to a Class-2 CSR if the queue length exceeds t, regardless of how many (as long as not all) Class-2 CSRs are busy. The threshold t will be optimally selected among all possible fixed thresholds. This policy simplifies the state-dependent threshold policy by using a single fixed (i.e., state-independent) threshold, and is optimal for the case of two heterogeneous CSRs ($C=2$ and $S_1=S_2=1$).

• A *$p\mu$ policy* further simplifies the *$p\mu$-t policy* by routing a call to an available Class-2 CSR as soon as possible. That is, it sets $t=0$.

For comparison purposes, we also study the following policy, which does not use $p\mu$ as a factor in the routing decisions:

• A *random assignment policy* routes a call randomly to *any* available CSR. This is the policy often used by call centers that do not incorporate any $p\mu$ information into routing decisions.

Our numerical analysis includes 54 cases, which cover light (Cases 1–18), medium (Cases 19–36), and heavy (Cases 37–54) traffic situations. Of the 18 cases for each situation, we analyze when $p_1\mu_1$ and $p_2\mu_2$ are close (the first nine cases) and far apart (the next nine cases). Then, for each fixed $p_i\mu_i$, $i=1,2$, we let p_i and μ_i take on three sets of values so that there are nine combinations. The purpose is to test "normal" cases as well as "extreme" cases, which will give us a sense of the "bound" on the differences. Detailed parameter values are given in Table 1. For each case, we compare the random assignment policy, the $p\mu$ policy, and the $p\mu$-t policy with the optimal state-dependent threshold policy determined numerically by a value iteration algorithm.

Results in Table 1 show that the benefit of allowing the threshold to vary state by state (i.e., optimal versus $p\mu$-t) is minimal. This is intuitive: Although the thresholds used by the (optimal) state-dependent threshold policy vary significantly between $n_2=0$ and $n_2=S_2-1$, only a few of these thresholds really matter, because most of the (S_1, n_2) states are visited very infrequently (if at all) in the steady state. Therefore, the $p\mu$-t policy, which uses the best t for all states, performs well. This also simplifies the search for optimal control parameters.

Furthermore, we observe that the benefit of withholding some calls (i.e., $p\mu$-t versus $p\mu$), similar to the benefit gained in the slow-server problem, is far less than the benefit of recognizing and utilizing the $p\mu$ rule in call routing (i.e., $p\mu$ versus random). Because the $p\mu$ policy does not require any computation except for the ranking of the $p\mu$ index, this means that in most cases the $p\mu$ policy is a better policy for implementation. Actually, the performance of the $p\mu$ policy is dramatically worse than that of the $p\mu$-t policy only for Cases 28, 31, and 34. These cases correspond to (1) medium-traffic situations (the utilization rate is 50%), (2) a wide difference between $p_1\mu_1$ and $p_2\mu_2$, and (3) $p_2=1$. To understand (1), we note that when traffic is high, Class-2 CSRs are heavily used and the optimal threshold is low. When traffic is low, Class-2 CSRs are hardly necessary. Both of these situations lead to a small difference between $p\mu$ and $p\mu$-t policies. For (2), when the CSR heterogeneity is higher, the optimal threshold should be higher, leading to a greater difference between $p\mu$ and $p\mu$-t policies. To see (3), we note that when $p_2<1$, an unresolved call by a Class-2 CSR can be rerouted to a Class-1 CSR. When $p_2=1$, however, once a call is routed to a Class-2 CSR, it remains there. Therefore, the use of a threshold to withhold calls becomes more important when $p_2=1$, leading to a greater difference between $p\mu$ and

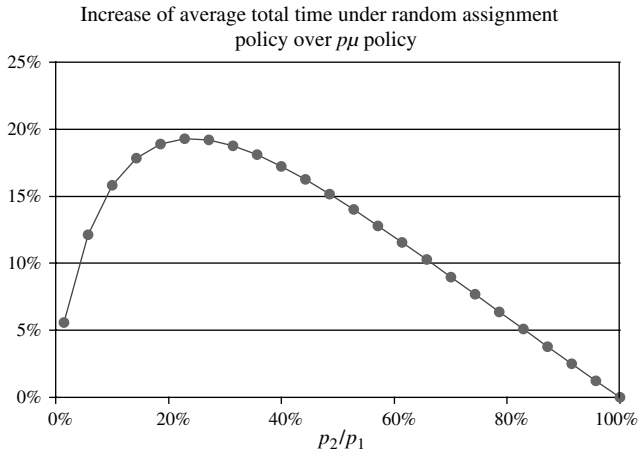
Table 1. Comparison of policies when $\lambda = 2$ and $S_1 = S_2 = 8$.

						Cost increase over optimal policy				
Case	p_1	μ_1	p_2	μ_2	ρ	Random assignment (%)	$p\mu$ (%)	$p\mu$-t (%)	Preemptive (%)	Dedicated (%)
1	0.65	1	1	0.6	0.2	4.74	0	0	−0.06	0.07
2	0.65	1	0.6	1	0.2	3.95	0	0	−0.04	0.09
3	0.65	1	0.06	10	0.2	1.02	0	0	0	0.12
4	1	0.65	1	0.6	0.2	4.05	0	0	−0.06	0.07
5	1	0.65	0.6	1	0.2	3.26	0	0	−0.04	0.09
6	1	0.65	0.06	10	0.2	0.73	0	0	0	0.12
7	0.1	6.5	1	0.6	0.2	7.09	0	0	−0.06	0.07
8	0.1	6.5	0.6	1	0.2	6.63	0	0	−0.04	0.09
9	0.1	6.5	0.06	10	0.2	3.29	0	0	0	0.12
10	0.95	1	1	0.3	0.2	94.07	0.23	0	−0.01	0
11	0.95	1	0.3	1	0.2	52.45	0.06	0	−0.01	0
12	0.95	1	0.03	10	0.2	8.95	0	0	−0.01	0
13	0.5	1.9	1	0.3	0.2	116.68	0.23	0	−0.01	0
14	0.5	1.9	0.3	1	0.2	73.8	0.06	0	−0.01	0
15	0.5	1.9	0.03	10	0.2	15.59	0	0	−0.01	0
16	0.1	9.5	1	0.3	0.2	161.45	0.23	0	−0.01	0
17	0.1	9.5	0.3	1	0.2	130.51	0.06	0	−0.01	0
18	0.1	9.5	0.03	10	0.2	50.45	0	0	−0.01	0
19	0.54	0.5	1	0.23	0.5	5.76	0	0	−1.62	1.31
20	0.54	0.5	0.46	0.5	0.5	5.07	0	0	−0.97	1.97
21	0.54	0.5	0.1	2.3	0.5	3.02	0	0	−0.3	2.67
22	1	0.27	1	0.23	0.5	4.68	0	0	−1.62	1.31
23	1	0.27	0.46	0.5	0.5	3.96	0	0	−0.97	1.97
24	1	0.27	0.1	2.3	0.5	2.18	0	0	−0.3	2.67
25	0.1	2.7	1	0.23	0.5	8.31	0	0	−1.62	1.31
26	0.1	2.7	0.46	0.5	0.5	8.05	0	0	−0.97	1.97
27	0.1	2.7	0.1	2.3	0.5	6.09	0	0	−0.3	2.67
28	0.8	0.5	1	0.1	0.5	82.41	14.49	0	−2.36	0.01
29	0.8	0.5	0.2	0.5	0.5	54.42	2.96	0.02	−1.7	0.68
30	0.8	0.5	0.02	5	0.5	15.09	0.01	0.01	−0.61	1.8
31	0.4	1	1	0.1	0.5	90.77	14.49	0	−2.36	0.01
32	0.4	1	0.2	0.5	0.5	68.11	2.96	0.02	−1.7	0.68
33	0.4	1	0.02	5	0.5	23.93	0.01	0.01	−0.61	1.8
34	0.1	4	1	0.1	0.5	101.29	14.49	0	−2.36	0.01
35	0.1	4	0.2	0.5	0.5	90.48	2.96	0.02	−1.7	0.68
36	0.1	4	0.02	5	0.5	49.92	0.01	0.01	−0.61	1.8
37	0.5	0.3	1	0.135	0.88	0.73	0	0	−0.5	2.23
38	0.5	0.3	0.45	0.3	0.88	0.7	0	0	−0.32	2.41
39	0.5	0.3	0.1	1.35	0.88	0.45	0	0	−0.11	2.64
40	1	0.15	1	0.135	0.88	0.55	0	0	−0.5	2.23
41	1	0.15	0.45	0.3	0.88	0.51	0	0	−0.32	2.41
42	1	0.15	0.1	1.35	0.88	0.3	0	0	−0.11	2.64
43	0.1	1.5	1	0.135	0.88	1.13	0	0	−0.5	2.23
44	0.1	1.5	0.45	0.3	0.88	1.17	0	0	−0.32	2.41
45	0.1	1.5	0.1	1.35	0.88	0.95	0	0	−0.11	2.64
46	0.7	0.3	1	0.075	0.88	7.65	1.58	0.01	−8.68	0.05
47	0.7	0.3	0.25	0.3	0.88	8.63	0.48	0.01	−5.22	3.84
48	0.7	0.3	0.03	2.5	0.88	5.32	0	0	−1.24	8.21
49	0.3	0.7	1	0.075	0.88	8.85	1.58	0.01	−8.68	0.05
50	0.3	0.7	0.25	0.3	0.88	10.74	0.48	0.01	−5.22	3.84
51	0.3	0.7	0.03	2.5	0.88	7.98	0	0	−1.24	8.21
52	0.1	2.1	1	0.075	0.88	9.91	1.58	0.01	−8.68	0.05
53	0.1	2.1	0.25	0.3	0.88	12.88	0.48	0.01	−5.22	3.84
54	0.1	2.1	0.03	2.5	0.88	12	0	0	−1.24	8.21

Table 2. Impact of the size of the call center.

Scale factor	λ	Optimal threshold for $p\mu$-t policy	Cost increase over optimal policy: $p\mu$-t policy (%)	Cost increase over optimal policy: $p\mu$ policy (%)
1	16	2	0.00	0.62
2	32	2	0.00	0.89
4	64	3	0.01	0.88
6	96	3	0.04	0.83
8	128	4	0.02	0.76
10	160	4	0.02	0.71
12	192	5	0.30	0.93
14	224	5	0.00	0.01
16	256	6	0.04	0.33
18	288	6	0.02	0.60
20	320	6	0.03	0.58

Figure 2. Comparing the $p\mu$ and random assignment policies: $\mu_1 = \mu_2$.



$p\mu$-t policies. In practical situations, the traffic is usually high, and $p_2 < 1$. Therefore, the difference between the $p\mu$ and $p\mu$-t policies diminishes.

We conclude this section by analyzing the performances of the $p\mu$-t and $p\mu$ policies as the size of the call center increases. Tested cases and results are presented in Table 2. For all cases, we let $\mu_1 = \mu_2 = 2$, $p_1 = 1$, $p_2 = 0.5$, and increase λ and $S_1 = S_2$ by a scale factor varying from 1 to 20 such that $\rho = 2/3$.

As shown by Table 2, the $p\mu$-t policy always performs very well with an error less than or equal to 0.3%. Maybe more interesting is the efficiency of the $p\mu$ policy, which has an error less than 1%. These results suggest that our findings for small systems remain true for larger ones. We also observe that as the system size grows (with traffic intensity at a fixed value), the threshold also increases, but at a lower rate, and remains small relative to the total number of CSRs.

We note that there could be another way of testing the size effect. Instead of fixing the system utilization as we increase the size of the call center, we could also fix a certain service level (e.g., 5% delay probability). As the arrival rate increases, the size of the call center would increase in a way that follows the square-root staffing rule (e.g., see Borst et al. 2004). One difficulty is that these rules are not derived for the heterogeneous servers, callback loops, and priority rules that are essential in our model. It would be an interesting area for future research to see how the square-root staffing rule can be adapted to our model.

5.2. Importance of Call Resolution Probability p

We want to stress in this paper the importance of incorporating the call resolution probability p into the call-routing priority index. In this section, we set $\mu_1 = \mu_2$ and fix p_1. Then, we systematically decreased p_2, starting from $p_2 = p_1$. If the manager of a call center only measures the speed of its CSRs, then it will assume that all CSRs are the same. Therefore, a random assignment policy will be used. On the other hand, if the call center measures the call resolution probability p of each CSR, then it should route calls according to p (i.e., use the $p\mu$ and $p\mu$-t policies).

The parameters used in the tests are as follows: $\lambda = 1$, $S_1 = S_2 = 8$, $\mu_1 = \mu_2 = 0.18$, and $p_1 = 0.7$; p_2 varies. Results are summarized in Figure 2.

We observe that the random policy performs very poorly against the $p\mu$ policy in most cases. In general, the smaller the ratio p_2/p_1, the bigger the difference. This is intuitive because the benefit of recognizing the difference between p_1 and p_2 and utilizing it in routing is greater when the difference is bigger. However, in the extreme, as p_2 approaches 0, the system traffic intensity approaches 0.99. This is very heavy traffic, and all the policies tend to use the Class-2 CSRs whenever possible. That explains why the difference narrows as $p_2 \to 0$.

Note that when $\mu_1 = \mu_2$, the $p\mu$ policy simply gives calls to the CSR class with the higher p. When the ranking of the CSRs according to their call resolution probabilities is common knowledge (or can be measured), such a policy is very easy to implement, and also gives significant benefits.

So far we have focused on $C = 2$. When $C \geqslant 3$, the optimal policy is complex because the $p\mu$ rule may not be optimal, and the thresholds may be state dependent. Even the $p\mu$-t policy may be too complex to implement because we need to find $C - 1$ fixed thresholds—one for each Class-i, $2 \leqslant i \leqslant C$. A detailed discussion of this is beyond the scope of this paper (see de Véricourt and Zhou 2004 for more details).

5.3. Dedicated Policy

So far, we have assumed that the service rates do not depend on the number of previous attempts to solve the customer's problem. In practical situations, the service time

may decrease if the customer talks to the same CSR (e.g., call centers dealing with complex issues such as medical and legal help). In such cases, it may indeed be better to route a callback to the CSR who answered this call the first time (the *original CSR*).

In this section, we numerically evaluate the performance of a *dedicated policy*. This policy allocates new calls according to the $p\mu$-t routing policy, but always routes callbacks immediately to the original CSR. The dedicated policy applies to situations where it is the CSR who reaches the conclusion that the problem has not been properly addressed. Instead of handing the call off to another CSR, which would result in another setup time, the CSR may want to keep the call and give it another try. The dedicated policy also applies to call centers requiring callbacks to enter a case number at the phone prompt that corresponds to the particular customer issue. For call centers that cannot identify the reason of a call as it enters the system, the implementation of the dedicated policy is difficult.

Let us assume that each time a callback is routed to the original CSR, the average service rate increases by a given percentage δ. In other words, the average service rate for the kth attempt is equal to $(1+\delta)^{k-1}\mu_i$ for a Class-i CSR. Therefore, the total average service time (taking the callbacks into account), $\tilde{\mu}_i$, is equal to $\mu_i(p_i+\delta)/(1+\delta)$. To simplify the analysis, we assume that the total service time is exponentially distributed, and the system becomes a slow-server problem with rates $\tilde{\mu}_i$.

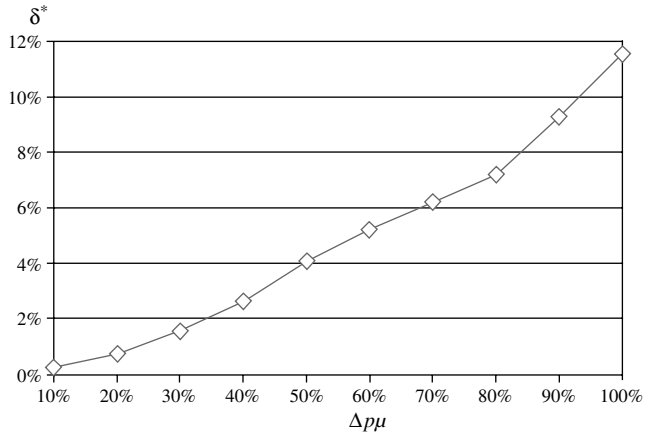
We compare the dedicated policy with the $p\mu$-t policy. We assume that the $p\mu$-t policy does not utilize customer callback information, so that it is unlikely for a callback to be reassigned to the original CSR. Therefore, we assume that under the $p\mu$-t policy the service rates do not depend on the number of previous attempts. At the end of this section, we discuss how to use callback information in the $p\mu$-t policy.

When $\delta = 0$, the total service time by a Class-i CSR is a geometric sum of exponential random times with the same rate μ_i, and the system is equivalent to a slow-server system with service rates of $\{p_i\mu_i\}$, and no callbacks. As δ increases (i.e., the time saving becomes larger), the gap between the dedicated policy and the $p\mu$-t should narrow. Eventually, there should exist a δ^* such that the dedicated policy outperforms the $p\mu$-t policy if $\delta \geqslant \delta^*$.

We let $S_1 = S_2 = 8$, $p_2\mu_2 = 1$, and let $p_1\mu_1 = \mu_2$ vary from 1.1 to 2. Figure 3 depicts δ^* as $\Delta p\mu := (p_1\mu_1 - p_2\mu_2)/p_2\mu_2$ increases. Although δ^* is increasing in $\Delta p\mu$, for δ^* to be significant $\Delta p\mu$ needs to be large. For instance, when $p_1\mu_1$ is 50% larger than $p_2\mu_2$, the dedicated routing policy should be used as soon as the CSRs can improve the service rates at each attempt by 4%.

This suggests that the dedicated routing policy should work well when $\Delta p\mu$ is not particularly large. It also suggests that when the original CSR of a callback can be identified, the call-routing policy should use this information. Here we propose a modified $p\mu$-t policy that takes

Figure 3. δ^* as a function of $\Delta p\mu$.



advantage of the benefits of the dedicated policy: The $p\mu$-t policy still determines when (i.e., when the queue length exceeds the threshold) and where (i.e., $p\mu$ rule) to route a call as before. In addition, if the call is a callback, the original CSR is idle and is in the class identified by the $p\mu$ rule, then the modified $p\mu$-t policy should route the call to the original CSR. Note that if the queue discipline is FCFS, the original CSR may not always be available. Even when callbacks are given priority in routing, they may still wait in the queue if, upon their return, all Class-1 CSRs are busy and the queue length is below threshold. By the time the callbacks are routed, their original CSRs may not be available either. In summary, the modified policy is based on the $p\mu$-t policy, but it routes the callbacks to their original CSR whenever possible. Other modifications of the $p\mu$-t are also possible. We believe that when δ is significant, these modified $p\mu$-t policies constitute good alternatives to the dedicated routing policy.

5.4. Delay in Callback

So far, we have assumed that when a call is not resolved successfully, the call returns immediately to the system. In many instances, however, the resolution of a call may not be immediate. Therefore, the customer leaves the system after being served, and calls back (if needed) only after a certain amount of time. From a modeling perspective, this system can be viewed as having a callback "orbit." Unresolved calls stay in the orbit for an exponentially distributed time with rate ν before coming back to the system. Because the number of calls in the orbit is usually unknown to the call center, this model is a partially observed MDP, for which general results and algorithms are limited (e.g., see Puterman 1994).

Let us call the immediate-callback model in §3 the IC model, and the delayed-callback model the DC model. Note that the IC model corresponds to the DC model in which $\nu = \infty$. In this section, we test how well the $p\mu$ heuristics (developed for the IC model) perform in the DC model. More precisely, we identify the best threshold of the $p\mu$-t

Table 3. Effect of ν for different μ_2 with $\lambda = 4$, $p_1 = 1$, $\mu_1 = 1$, $p_2 = 0.6$, and $S_1 = S_2 = 4$.

			Cost increase over lower bound	
Case	μ_2	ν	$p\mu$-t policy (%)	$p\mu$ policy (%)
1	1	∞	0.0	0.0
2	1	1	0.0	0.0
3	1	1/2	0.0	0.0
4	1	1/4	0.0	0.0
5	1	1/8	0.0	0.0
6	1	1/16	0.0	0.0
7	1	1/32	0.0	0.0
8	1	1/64	0.0	0.0
9	0.8	∞	0.1	0.1
10	0.8	1	0.2	0.3
11	0.8	1/2	0.1	0.4
12	0.8	1/4	0.1	0.4
13	0.8	1/8	0.0	0.5
14	0.8	1/16	0.0	0.5
15	0.8	1/32	0.0	0.5
16	0.8	1/64	0.0	0.5
17	1.2	∞	0.0	0.0
18	1.2	1	0.0	0.0
19	1.2	1/2	0.0	0.0
20	1.2	1/4	0.0	0.0
21	1.2	1/8	0.0	0.0
22	1.2	1/16	0.0	0.0
23	1.2	1/32	0.0	0.0
24	1.2	1/64	0.0	0.0

policy for the IC model and apply the same policy to the corresponding DC model. We also test the $p\mu$ policy.

Because the DC model cannot be evaluated, we actually look at another system in which the orbit size is limited and the full state information, including the size of the callback orbit, is known to the decision maker. Both assumptions reduce the system cost, resulting in a lower bound of the system. It is against this lower bound that we numerically test the performance of the $p\mu$-t and $p\mu$ policies (see the online appendix for more details).

Tests for 24 cases are summarized in Table 3. In Cases 1–8, 9–16, and 17–24, we have $\mu_1 = \mu_2$, $\mu_1 > \mu_2$, and $\mu_1 < \mu_2$, respectively. We also start with the IC model ($\nu = \infty$). Then, we gradually decrease ν to 1/64. If an average call lasts five minutes, then $\nu = 1/64$ corresponds to an average delay of more than five hours before callback. Values for the other parameter values and the results are given in Table 3.

Results in Table 3 suggest that further decreasing ν will not have a significant impact. Moreover,

- Both the IC $p\mu$-t and $p\mu$ policies, when applied to the DC model, have costs that are extremely close to the DC lower bound (all errors are less than 1%). This suggests that in practice, the information about the orbit size (i.e., the number of unresolved calls that will eventually come back) is not necessary.
- As the average delay of callback varies from "immediate" to "more than five hours," no significant changes are noted, providing another justification for the immediate-callback assumption: The $p\mu$-t and $p\mu$ policies generated by the IC model work very well in the DC model, where there is a significant delay in the callback.

An explanation for the insensitivity of the results to the delay is that in the MDP formulation, we study the steady-state behavior of a stationary queueing system. With or without delay, unresolved calls will eventually come back. Therefore, the delay orbit changes the timing of the callbacks, but very little of the rate of the callbacks. Consequently, the total rate of callbacks to the system is similar for both the IC and the DC models. In the IC model, there is a strong correlation between service completion and callback arrivals. In the DC model, due to the (especially exponential) delay between the two events, the correlation becomes weaker. Numerical results in Table 3 suggests that the similarity of overall callback rate between the IC and DC models has a much stronger effect than the difference between the two models caused by the completion-callback correlation.

5.5. Lower-Bound Policy

The $p\mu$, $p\mu$-t, and dedicated policies (see Table 1, Column 11 for the performance of the dedicated policy when $\delta = 0$) perform well in general, but they all provide an upper bound on the performance of the optimal policy studied in §3. To complete the analysis, we provide in this section the closed-form solution to a policy that gives a lower bound on the optimal system performance.

The lower-bound policy we study is the *preemptive policy*: At any time, even if a call is already being served by a CSR, we allow it to be handed over to another CSR during the service. Because the service times are exponentially distributed, we can assume that the call starts over after the hand-off. The preemptive assumption is very restrictive, making the policy applicable only to call centers where customers tolerate such hand-offs. Nevertheless, it provides a good lower bound that is also easy to evaluate.

Because preemption is allowed at any time, it makes sense to not hold calls in the queue when there are idle CSRs (one can always reroute the calls later). Moreover, a call should always be routed or rerouted to the highest-$p\mu$ available CSR. These intuitions are formalized by the following theorem.

THEOREM 4. *The optimal preemptive policy always routes* (*or reroutes, if the call is already in service*) *a call to the available CSR with the highest $p\mu$ index.*

PROOF. The proof uses a coupling argument, and can be found in the online appendix.

As a result of Theorem 4, when a service is completed and the customer is dissatisfied, the call would be routed

to the same CSR (otherwise, this call would have been rerouted earlier). Therefore, without loss of generality, we can set $p_1 = p_2 = \cdots = p_C = 1$, and μ_i to be the original $p_i\mu_i$ in the following analysis.

Due to preemption, calls in the system will always be handled by the fastest CSRs. For example, if there are i calls in the system, where $S_1 < i \leqslant S_1 + S_2$, then S_1 of the calls will be handled by Class-1 CSRs and $i - S_1$ of them will be handled by Class-2 CSRs. As a result, the only variable we need to keep track of is the total number of calls in the system, i. Therefore, if μ_i denotes the total rate of service completion in state i, then

$$\mu_i = \begin{cases} i\mu_1 & \text{for } 0 \leqslant i \leqslant S_1, \\ S_1\mu_1 + (i - S_1)\mu_2 & \text{for } S_1 < i \leqslant S_1 + S_2, \\ \cdots & \\ \displaystyle\sum_{k=1}^{C-1} S_k\mu_k + \left(i - \sum_{k=1}^{C-1} S_k\right)\mu_C & \\ & \text{for } \displaystyle\sum_{k=1}^{C-1} S_k < i \leqslant \sum_{k=1}^{C} S_k, \\ \displaystyle\sum_{k=1}^{C} S_k\mu_k & \text{for } \displaystyle\sum_{k=1}^{C} S_k < i. \end{cases} \tag{13}$$

If we let q_i denote the steady-state probability of the system being in state i, then the state-transition balance equations are $\lambda q_i = \mu_{i+1} q_{i+1}\ \forall i$.

Therefore, if we let $S = \sum_{k=1}^{C} S_k$ and $\mu_S = \sum_{k=1}^{C} S_k\mu_k$, we must have

$$q_i = \begin{cases} q_S \cdot \displaystyle\prod_{j=i+1}^{S} \left(\frac{\mu_j}{\lambda}\right) & \forall\, 0 \leqslant i \leqslant S-1, \\ q_S \cdot \left(\dfrac{\lambda}{\mu_S}\right)^{i-S} & \forall\, i \geqslant S. \end{cases} \tag{14}$$

Solving the probability uniformization equation $\sum_{i=0}^{\infty} p_i = 1$, we obtain

$$q_S = \frac{1}{\sum_{i=0}^{S-1}\left[\prod_{j=i+1}^{S}(\mu_j/\lambda)\right] + 1/(1 - (\lambda/\mu_S))}. \tag{15}$$

This, along with (14), uniquely determine all of the steady-state probabilities. We can use these steady-state probabilities to calculate the average number in the system, which will give us a lower bound on the performance of our system:

$$L_s = q_S\left[\sum_{i=1}^{S-1}\left(i\prod_{j=i+1}^{S}\frac{\mu_j}{\lambda}\right) + \frac{\lambda\mu_S}{(\mu_S - \lambda)^2} + \frac{S\mu_S}{\mu_S - \lambda}\right]. \tag{16}$$

Using the closed-form expressions given in (14)–(16), we can quickly compute the performance of this preemptive system. To see how tight the lower bound is, we test it using the 54 cases in §5.1. Results are included in Table 1. For most cases, the lower bound is within 2.5% of the optimal policy. The cases with bad performance (46, 47, 49, 50, 52, and 53) are extreme cases in which $\mu_1 \gg \mu_2$. They are very unlikely to occur in practice.

6. Conclusion

Traditional research on routing decisions focuses on speed and waiting cost. Service quality related metrics are rarely taken into account for such operational decisions, although they play a crucial role in the short-term traffic reduction and long-term customer loyalty of a firm. We see our research as a promising step in showing that service quality can be—and should be—incorporated into operational decisions.

In this paper, we consider both service speed and quality in routing decisions for a telephone call center. We argue that call resolution probability p is a good measure of call quality. An MDP model is used to characterize the optimal routing policy. Our main contribution is to identify the call resolution rate $p\mu$ as a simple priority index in routing calls: First, we show that the use of the $p\mu$ rule is optimal in a broad set of cases. Then, we show that the $p\mu$-threshold policies are optimal in certain cases. Finally, we show numerically that simple $p\mu$-based policies work well as heuristics. These numerical tests highlight the benefits that can be achieved by considering p, in addition to the traditional measure of μ, when making routing decisions.

Even though results in this paper focus primarily on the short-term benefit of traffic reduction, incorporating quality-related metrics into routing decisions could also provide significant long-term benefits. For instance, to achieve the same service level on customer waiting time, fewer low-$p\mu$ CSRs are needed under a $p\mu$-based policy than under a μ-based policy. The freed-up low-$p\mu$ CSRs can then be scheduled to receive training. Over the long run, the call center could improve its CSRs' service speed and/or quality, all without adding extra personnel or sacrificing service level. For companies in the process of migrating from the traditional cost-based metrics (e.g., average wait time) to the profit-based metrics (e.g., call resolution probability), our procedure helps them to maintain the service measured by current metrics in the short run, while increasing their service as measured by new metrics in the long run.

In de Véricourt and Zhou (2004), we use a numerical example to illustrate this. We scale down a real call center's call arrival data for every 30 minutes for a full day, and run a workforce-scheduling linear program to figure out the minimum staffing level to satisfy a given service level for each 30-minute interval. We show that the random assignment policy will schedule ten Class-1 and nine Class-2 CSRs. By using a $p\mu$-based policy, we can achieve the same (or better) service level for every 30-minute interval, with only ten Class-1 and six Class-2 CSRs. This means full-day training of three Class-2 CSRs can be achieved, which is clearly a significant improvement.

In our future research, we will examine other long-term benefits such as customer loyalty. For example, when a customer is dissatisfied with a service, he or she may simply defect and never call back. Therefore, if call quality is not carefully considered in routing decisions, a company could

lose many customers in the long run due to poor service quality. For call centers that outsource, this also has an impact on how both speed and resolution probability should be specified in contracts.

Appendix

Please see the online companion to this paper at http://or.pubs.informs.org/Pages/collect.html.

Acknowledgments

The authors thank Paul Zipkin for his insightful comments on an earlier version of this paper. They also thank the seminar participants at Duke University and the University of North Carolina for their helpful suggestions. The many helpful comments and suggestions by two anonymous reviewers and the associate editor have significantly improved the paper, and they are much appreciated.

References

Atar, R., A. Mandelbaum, M. Reiman. 2004. Scheduling a multi-class queue with many exponential servers: Asymptotic optimality in heavy-traffic. *Ann. Appl. Probab.* **14**(3) 1084–1134.

Bell, S. L., R. J. Williams. 2001. Dynamic scheduling of a system with two parallel servers in heavy traffic with complete resource pooling: Asymptotic optimality of a continuous review threshold policy. *Ann. Appl. Probab.* **11** 608–649.

Berk, E., K. Moinzadeh. 1998. The impact of discharge decisions on health care quality. *Management Sci.* **44**(3) 400–415.

Borst, S., A. Mandelbaum, M. Reiman. 2004. Dimensioning large call centers. *Oper. Res.* **52**(1) 17–34.

De Angelis, V. 1998. Planning home assistance for AIDS patients in the city of Rome, Italy. *Interfaces* **28**(3) 75–83.

de Véricourt, F., Y.-P. Zhou. 2004. Managing service time and service failure in a call routing problem. Working paper, University of Washington, Seattle, WA.

de Véricourt, F., Y.-P. Zhou. 2005. On the incomplete results for the multiple-server slow-server problem. Technical report, Duke University, Durham, NC.

Falin, G. I., J. G. C. Templeton. 1997. *Retrial Queues*. Chapman & Hall, London, UK.

Gans, N. 2002. Customer loyalty and supplier quality competition. *Management Sci.* **48**(2) 207–221.

Gans, N., Y.-P. Zhou. 2002. Managing learning and turnover in employee staffing. *Oper. Res.* **50**(6) 991–1006.

Gans, N., Y.-P. Zhou. 2003. A call-routing problem with service-level constraints. *Oper. Res.* **51**(2) 255–271.

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

Ha, A. 1997. Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Res. Logist.* **44** 457–472.

Hall, J., E. Porteus. 2000. Customer service competition in capacitated systems. *Manufacturing Service Oper. Management* **2**(2) 144–165.

Harrison, J. M., M. J. Lopéz. 1999. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems* **33**(4) 339–368.

Heskett, J., T. Jones, G. Loveman, W. E. Sasser Jr., L. Schlesinger. 1994. Putting the service-profit chain to work. *Harvard Bus. Rev.* (March–April) 164–174.

Koole, G. 1995. A simple proof of the optimality of a threshold policy in a two-server queueing system. *Systems Control Lett.* **26**(5) 301–303.

Larsen, R., A. K. Agrawala. 1983. Control of a heterogeneous two-server exponential queueing system. *IEEE Trans. Software Engrg.* **9**(4) 522–526.

Larsen, R. L. 1981. Control of multiple exponential servers with application to computer systems. Ph.D. dissertation, Department of Computer Science, University of Maryland, College Park, MD.

Lin, W., P. R. Kumar. 1984. Optimal control of a queueing system with two heterogeneous servers. *IEEE Trans. Automatic Control* **AC-29**(8).

Luh, H., I. Viniotis. 2002. Threshold control policies for heterogeneous server systems. *Math. Methods Oper. Res.* **55**(1) 121–142.

Mandelbaum, A., A. Stolyar. 2002. $Gc\mu$ scheduling of flexible servers: Asymptotic optimality in heavy traffic. Technical report, Technion, Haifa, Israel.

Mandelbaum, A., W. A. Massey, M. Reiman, B. Rider. 1999a. Time varying multiserver queues with abandonment and retrials. P. Key, D. Smith, eds. *Teletraffic Engineering in a Competitive World*, ITC-16. Elsevier, 355–364.

Mandelbaum, A., W. A. Massey, M. Reiman, A. Stolyar. 1999b. Waiting time asymptotics for time varying multiserver queues with abandonment and retrials. *Thirty-Seventh Annual Allerton Conference*, 1095–1104.

Mandelbaum, A., W. A. Massey, M. Reiman, B. Rider, A. Stolyar. 2000. Queue lengths and waiting times for multiserver queues with abandonment and retrials. *Selected Proc. Fifth INFORMS Telecomm. Conf.*

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York.

Read, B. 2002. The struggling revolution. *Call Center Magazine* (Dec. 4). Downloadable from http://www.callcentermagazine.com/article/CCM20021202S0005.

Rykov, V. V. 2001. Monotone control of queueing systems with heterogeneous servers. *Queueing Systems* **37**(4) 391–403.

Shumsky, R., E. Pinker. 2003. Gatekeepers and referrals in services. *Management Sci.* **49**(7) 839–856.

Teh, Y.-C., A. R. Ward. 2002. Critical thresholds for dynamic routing in queueing networks. *Queueing Systems* **42**(3) 297–316.

Van Mieghem, J. A. 1995. Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *Ann. Appl. Probab.* **5**(3) 809–833.

Veatch, M., L. Wein. 1996. Scheduling a make-to-stock queue: Index policies and hedging points. *Oper. Res.* **44**(4) 634–647.

Walrand, J. 1984. A note on "Optimal control of a queueing system with two heterogeneous servers." *Systems Control Lett.* **4**(3) 131–134.

Zeithaml, V., A. Parasuraman, L. Berry. 1993. The nature and determinants of customer expectations of service. *Acad. Marketing Sci.* **21**(1) 1–12.