Manufacturing & Service Operations Management

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

Threshold Routing to Trade Off Waiting and Call Resolution in Call Centers

Dongyuan Zhan, Amy R. Ward

M&SOM MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 16, No. 2, Spring 2014, pp. 220–237
ISSN 1523-4614 (print) | ISSN 1526-5498 (online)

informs®
http://dx.doi.org/10.1287/msom.2013.0463

Threshold Routing to Trade Off Waiting and Call Resolution in Call Centers

Dongyuan Zhan, Amy R. Ward
Marshall School of Business, University of Southern California, Los Angeles, California 90089
{dongyuan.zhan.2015@marshall.usc.edu, amy.ward@marshall.usc.edu}

In a call center, agents may handle calls at different speeds, and also may be more or less successful at resolving customers' inquiries, even when only considering customers calling with similar requests. One common measure of successful call resolution is whether or not the call results in the customer calling back. This presents a natural trade-off between speed and quality, where speed is defined as the average time before an incoming call is answered (the average waiting time) and quality is defined as the percentage of all arriving calls that do not result in callbacks (the call resolution). The relevant control is the routing, that is, the decision concerning which agent should handle an arriving call when more than one agent is available. In an inverted-V model setting, we formulate an optimization problem with the dual performance objective of minimizing average customer waiting time and maximizing the call resolution. We solve this optimization problem asymptotically in the Halfin–Whitt many-server limit regime, interpret its solution as a routing control for the discrete-event system, and show via simulation that the interpreted routing control is on the efficient frontier. In particular, any routing control that has a lower average waiting time (higher call resolution) must also have a lower call resolution (higher average waiting time).

Keywords: call center management; queueing theory; stochastic methods; service operations
History: Received: August 3, 2012; accepted: August 2, 2013. Published online in *Articles in Advance* November 20, 2013.

1. Introduction

Speed and quality are two key measures to evaluate the operational performance of a manufacturer or a service provider. In general, these two measures are not independent. For example, it is often the case that the quality of a good or service degrades as the production or service speed increases. In other words, there is a trade-off between quality and speed.

In a call center environment, agent heterogeneity drives the speed–quality trade-off. One measure of agent heterogeneity is the average time an agent spends talking to each customer, which determines the service speed. Another measure is an agent's resolution probability, that is, the percentage of calls successfully handled by that agent that do not result in a callback (a follow-up call to the call center for the same problem, because that agent did not adequately answer the customer's question). The callback is a sign of poor service quality, because it indicates that the agent did not properly answer the customer's question. Sometimes, agents with slow service speed also have low resolution probability, because they are inexperienced. Then, there is no conflict between speed and quality when routing arriving customers to available agents. However, sometimes, agents with slow service speed have high resolution probability. This is because slow service speed can result from the agents spending a longer time listening to and interacting with their customers. Then, there is the following speed–quality trade-off when routing arriving customers to available agents: should the agent with faster speed or with higher resolution probability serve the customer?

To answer this question, it is necessary to know the performance objective. One primary objective in call center management is to minimize the steady-state average waiting time. Then, the routing rule that ranks agents according to their effective service rate (their service rate times their resolution probability) and sends an arriving customer to the available agent with the highest effective service rate has been shown to perform very well (for an exact analysis in an inverted-V model that includes callbacks, see de Vericourt and Zhou 2005; for an asymptotic analysis, see Armony 2005). However, Mehrotra et al. (2012) show that such a routing rule may perform very poorly with respect to objectives that involve the call resolution (i.e., the overall percentage of arriving calls that do not result in a callback), and the importance of such objectives is recognized in Hart et al. (2006). For example, if the system manager wishes to maximize the call resolution, then the routing rule that ranks agents according to their resolution probability is likely to outperform the aforementioned rule that ranks agents according to their effective service rates. Note that we differentiate between an agent's *resolution probability* (the probability that *that* agent successfully resolves a call) and the overall system *call resolution* (the probability that an arbitrary arriving call is successfully resolved).

Our objective is to develop routing rules that take into account the dual performance objectives of minimizing the steady-state average waiting time and maximizing the steady-state call resolution. More specifically, we would like to find a routing rule that lies on the efficient frontier with respect to these two performance measures; that is, any routing rule that has a lower average waiting time (higher call resolution) than ours must also have a lower call resolution (higher average waiting time).

Our ideal is to consider this objective in the context of a call center that has customers calling with different types of requests and agents that are heterogeneous with respect to their service speeds and resolution probabilities, even within the context of the same call type (as is true in the call center data set in Mehrotra et al. 2012). However, finding an analytic solution to that problem is very ambitious (as evidenced by the fact that the results in Mehrotra et al. 2012 for call centers with heterogeneous customers and heterogeneous agents are all attained via carefully designed simulation studies). Fortunately, the simpler problem that assumes homogeneous customers and heterogeneous agents can be viewed as a building block to solving the heterogeneous customer and heterogeneous agent problem. This is because the analysis in Gurvich and Whitt (2009b) and Ward and Armony (2013) for two different call center routing problems that involve heterogeneous customers and agents shows that for large call centers both respective problems separate into one for an inverted-V model (homogeneous customers and heterogeneous agents) and one for a V model (heterogeneous customers and homogeneous agents). More specifically, that separation is true in the many server quality and efficiency driven (QED) regime, first formally defined in Halfin and Whitt (1981). Therefore, we consider the aforementioned objective in the context of a call center that has homogeneous customers and agents that are heterogeneous with respect to their service speeds and call resolution probabilities (i.e., an inverted-V model with multiple agent pools). Furthermore, we assume that the call center operates in the QED regime. We note that the QED regime is a natural regime to consider because it is a regime in which average waiting times are small and agents are highly utilized and can arise as an economically optimal operating regime; see Borst et al. (2004).

For the inverted-V model, we solve the diffusion control problem (DCP) that approximates (in the QED regime) the optimization problem whose objective is to minimize a weighted sum of the steady-state average waiting time and the steady-state average callback rate. We argue that any solution to this problem has associated average waiting time and call resolution on the efficient frontier. The solution to the DCP shows that some pools should always have routing priority, and so can be considered "reduced." The remaining pools have state-dependent priorities that are determined using a threshold control. The DCP solution translates into our proposed routing control, the reduced pools threshold (RPT) control. Finally, we use simulation to evaluate the performance of our proposed RPT control.

We complete this introduction with a brief review of the most relevant literature. Then, in §2, we present our model and problem formulation. In §3, we formulate and solve the approximating DCP. In §4, we propose the RPT control resulting from the DCP solution. In §5, we evaluate the performance of the RPT control via simulation. In §6, we make concluding remarks and propose directions for future research. The proofs of all our results can be found in the electronic companion (EC; available as supplemental material at http://dx.doi.org/10.1287/msom.2013.0463), as well as more on the simulation study.

1.1. Literature Review

There is a large literature on call centers. We focus on the papers most relevant to ours, and we refer the reader to the survey papers Gans et al. (2003) and Aksin et al. (2007) for a broad review of the call center literature.

Our work is motivated by Mehrotra et al. (2012). Their very insightful observation is that focusing solely on routing calls to minimize average waiting time produces routing controls that ignore the fact that agent resolution probabilities are different. Their paper proposes several different call routing rules and performs comprehensive simulation work (using data obtained from real-life call center) to identify an efficient frontier. However, there is no analytic justification of the efficient frontier.

Unfortunately, performing an exact analysis to identify points located on the efficient frontier is prohibitively difficult. Therefore, we perform an asymptotic analysis in the Halfin–Whitt many-server QED limit regime. This approach is common in the call center literature. The most closely related papers are those which assume QED staffing and optimize system performance with respect to a given performance objective, such as Armony and Ward (2010), Atar (2005), Atar et al. (2004), Gurvich and Whitt (2009b), Harrison and Zeevi (2004), Tezcan and Dai (2010), and Weerasinghe and Mandelbaum (2013). However, none of these papers use callbacks as a performance objective.

Our ideal would have been to analyze the model of Mehrotra et al. (2012) in the Halfin–Whitt many-server QED limit regime and to identify points located on the efficient frontier analytically in that regime. However, the model of Mehrotra et al. (2012) has multiple call types and multiple agent pools, and the agent service speed depends on both the agent pool and the call type. Then, formulating and solving an approximating DCP is hard. Therefore, we restrict our model to a single customer type and many agent pools (inverted-V model), which can be approximated by a one-dimensional DCP and so is analytically tractable. For work on models with multiple call types and multiple agent pools in which the agent service speed may depend on both the agent pool and the call type when there are no callbacks, we refer readers to Perry and Whitt (2009, 2011).

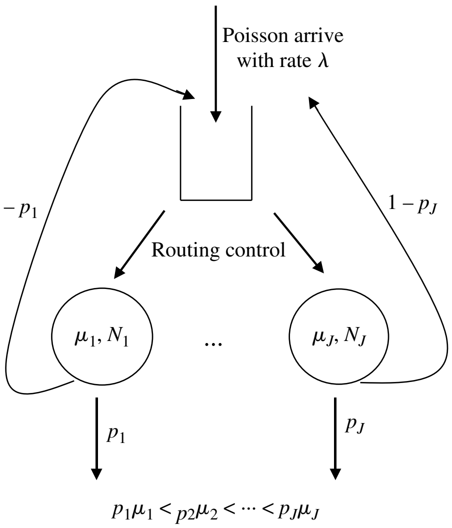
Our performance objective combines the dual goals of high speed (minimizing average waiting time) and good quality (maximizing call resolution). There are many papers that discuss speed–quality trade-offs. Recent work includes Alizamir et al. (2013), Anand et al. (2011), and Kostami and Rajagopalan (2014). However, there is no concept in those papers of poor service quality resulting in a follow-up service request (retrial), or a callback, in the call center terminology. This concept is key to our model, because we view the callback as an important measure of service quality.

Retrials appear in many other fields other than call centers. In the manufacturing setting, the models of Lovejoy and Sethuraman (2000) and Lu et al. (2009) both have speed–quality trade-offs and allow for rework. However, the first paper focuses on when to bring in overtime workers, and the second focuses on endogenized routing schemes, neither of which matches our focus. In health care, readmissions are akin to callbacks in call centers. The model of Chan et al. (2012) studies a similar speed–quality trade-off, specifically, a trade-off between the service rate and the probability of readmission to intensive care units. Their control is the service speed and their analysis is based on fluid approximation, whereas our control is routing, and our analysis is based on diffusion approximation.

2. The Model and Problem Formulation

Consider an inverted-V model with J pools and callbacks, as shown in Figure 1. The model is a parallel server system with a single customer class and J agent types, with each type in its own agent pool, all capable of fully handling customers' service requirements. Customers arrive to the system according to a Poisson process with rate λ and are served in the order of their arrival. Service times are independent and exponential, and the mean service time of a customer served by an agent from pool j is $1/\mu_j$, $j \in \mathcal{J} := \{1, 2, \ldots, J\}$. There is a probability $1 - p_j$, $j \in \mathcal{J}$, that a customer who completes service with a pool j agent has not had his or her request adequately handled. In that case, the customer immediately calls back, meaning he or she immediately returns to the system as an arriving customer for another independent service. We assume the available real-time information is unsophisticated and does not differentiate between customers calling for the first time and those calling back, even though such knowledge can be learned by later analysis of the recorded data (so that the agents can be separated into pools). Then, any customer calling back may be routed to the same pool as his previous service, or another pool, and the service rate and resolution probabilities are again the pool-dependent service rates and resolution probabilities μ_j and p_j, $j \in \mathcal{J}$. We let $\vec{\mu}$ and $\vec{p}$ represent the vectors of pool service rates and resolution probabilities.

Figure 1 **Inverted-V Call Center Model with Callbacks**



There are N_j agents in pool $j \in \mathcal{J}$, and we define $N := \sum_{j \in \mathcal{J}} N_j$. We assume

$$p_1\mu_1 < p_2\mu_2 < \cdots < p_J\mu_J, \tag{1}$$

meaning that the pools are labeled according to the order of their effective service rates $p_j\mu_j$.

Customers that arrive to the system when more than one agent pool has idle agents can be routed to any of the pools with idle agents. One natural routing control is the $p\mu$-rule, which routes customers to the available pool that has the highest effective service rate. Another natural control is the p-rule, which routes customers to the available pool that has the highest resolution probability. The $p\mu$-rule is an intuitive rule for minimizing the average waiting time, whereas the p-rule focuses on maximizing call resolution. The issue is that under the $p\mu$-rule the call resolution may be low, and under the p-rule the average waiting time may be high. In particular, there may be a trade-off between minimizing average waiting time and maximizing the call resolution. Our objective is to devise a routing control that optimally trades off these two competing objectives.

Denote by $\pi := \pi(\lambda, \vec{N})$ a routing control for the system with arrival rate λ and staffing vector $\vec{N} := (N_1, N_2, \ldots, N_J)$. Note that we omit the arguments λ and $\vec{N}$ when it is clear from the context which arguments should be used. Let $t \geq 0$ be an arbitrary time point. We denote by $Z_j(t;\pi)$ the number of busy agents in pool $j \in \mathcal{J}$ at time t, and denote by $I_j(t;\pi) := N_j - Z_j(t;\pi)$ the number of idle agents in pool $j \in \mathcal{J}$. Then, the instantaneous rate at which customers served by pool j agents call back is $(1-p_j)\mu_j Z_j(t;\pi)$. The number of customers waiting to be served at time t is $Q(t;\pi)$, and the total number of customers in the system is

$$X(t;\pi) := Q(t;\pi) + \sum_{j\in\mathcal{J}} Z_j(t;\pi).$$

The total number of idle agents is

$$I(t;\pi) := \sum_{j\in\mathcal{J}} I_j(t;\pi).$$

We omit the time argument when we refer to the entire process. We use $t = \infty$ whenever we refer to a process in steady state. Also, we omit π from the notation unless it is necessary to avoid confusion between different routing controls.

Let Π be the set of all nonanticipating, nonpreemptive, nonidling controls under which a unique steady state for X, Q, and Z_j, $j \in \mathcal{J}$ exists (which implies a unique steady state exists for I_j, $j \in \mathcal{J}$, and W as well). Nonanticipating (roughly speaking) means that a control cannot require knowledge of the future. By nonpreemptive, we mean that once a call is assigned to a particular agent, it cannot be transferred to another agent, nor can it be preempted by another call. Nonidling controls are those under which there can never simultaneously be waiting customers and idle agents.

We assume that the system load is less than one:

$$\rho := \frac{\lambda}{\sum_{j\in\mathcal{J}} p_j \mu_j N_j} < 1. \tag{2}$$

Then, the system is stable in the following sense.

Proposition 1. *When* (2) *holds, a unique steady state for X, Q, and Z_j, $j \in \mathcal{J}$, exists under any nonidling stationary Markovian control. Otherwise, there does not exist a steady state under any stationary Markovian control.*

For any $\pi \in \Pi$, the steady-state effective arrival rate to the system, inclusive of callbacks, is

$$\lambda_e(\infty;\pi) := \lambda + \sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\pi)],$$

and the steady-state call resolution is

$$p(\infty;\pi) := \frac{\lambda}{\lambda_e(\infty;\pi)}.$$

The total expected amount of time a customer spends waiting (including any time spent waiting after calling back) is

$$E[W_\Sigma(\infty;\pi)] = \frac{E[W(\infty;\pi)]}{p(\infty;\pi)},$$

where

$$E[W(\infty;\pi)] = \frac{E[Q(\infty;\pi)]}{\lambda_e(\infty;\pi)}$$

is the expected amount of time an arriving customer (new or callback) has to wait before reaching an agent.

Our objective is to find a routing control $\pi^\star \in \Pi$ whose associated average waiting time $E[W_\Sigma(\infty;\pi^\star)]$ and call resolution are on the efficient frontier. To do this, we let $c > 0$ and solve

$$\mathscr{C}_\star := \underset{\pi\in\Pi}{\text{minimize}}\; c\lambda E[W_\Sigma(\infty;\pi)] + \sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\pi)]. \tag{3}$$

Next, we observe that any routing control $\pi^\star \in \Pi$ that achieves the minimum in (3) for a given c is on the efficient frontier. To see this, suppose that $\tilde{\pi} \in \Pi$ is such that $E[W_\Sigma(\infty;\tilde{\pi})] < E[W_\Sigma(\infty;\pi^\star)]$. We claim it is also the case that

$$p(\infty;\tilde{\pi}) < p(\infty;\pi^\star). \tag{4}$$

To see this, first note that for any $\pi \in \Pi$,

$$\sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\pi)] = \frac{\lambda}{p(\infty;\pi)} - \lambda.$$

Then, if (4) does not hold, it must also be the case that

$$\sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\tilde{\pi})] \leq \sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\pi^\star)],$$

and so

$$c\lambda E[W_\Sigma(\infty;\tilde{\pi})] + \sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\tilde{\pi})] < c\lambda E[W_\Sigma(\infty;\pi^\star)] + \sum_{j\in\mathcal{J}} (1-p_j)\mu_j E[Z_j(\infty;\pi^\star)],$$

which contradicts the definition of $\pi^\star$. Therefore, if we can solve (3) for any $c \geq 0$, then by varying c from 0 to ∞, we produce a set of points on the efficient frontier.

Although proving the continuity of that set is much more difficult, looking ahead in the paper, our results for systems with large λ do support such continuity; see Remark 1 at the end of §3.

The problem (3) is a very complex Markov decision problem. Therefore, instead of solving (3) exactly, we formulate and solve the DCP that arises as an approximation to (3) in the Halfin–Whitt many-server limit regime (§3). We then use the solution to the approximating diffusion control problem to motivate a proposed routing control (§4), and we test the performance of that control via simulation (§5).

Before specifying the Halfin–Whitt limit regime, we interpret the parameter c in (3). We first observe that the objective function (3) is equivalently expressed in terms of the queue length as

$$\begin{aligned}\mathscr{C}_\star = \underset{\pi\in\Pi}{\text{minimize}}\ & cE[Q(\infty;\pi)] \\ & + \sum_{j\in\mathscr{J}}(1-p_j)\mu_j E[Z_j(\infty;\pi)]. \qquad (5)\end{aligned}$$

This follows from Little's Law, because

$$\lambda E[W_\Sigma(\infty;\pi)] = \lambda_e(\infty;\pi)E[W(\infty;\pi)] = E[Q(\infty;\pi)].$$

Then, the parameter c determines the relative cost of customers queueing in comparison with customers calling back.

Finally, it is worthwhile to comment on some of our modeling assumptions. The assumption of Poisson arrivals and exponential service times is common in the literature and is made for analytic tractability. However, the structure of the control we propose may still be good, even without these assumptions, and we perform a supporting numerical study in §5 (see Figure 8). Next, the assumption that callbacks are immediate is a simplification. This is reasonable for situations in which the customer can quickly assess whether or not the agent's answer is helpful. However, in general, callbacks are not immediate. Third, we have assumed that the service time and the fact that the call is a callback are independent. This is consistent with our assumption that the customers in the queue are not differentiated by whether or not they are callbacks. Last, our objective function assumes that callbacks are negative and should be minimized, which is reasonable in, for example, a call center that performs technical support. However, in the context of a call center that generates revenue (and probably also under the assumption that there may be delay before the callback occurs), callbacks can be positive, because they are an indication that the customer is returning to buy another item. Then, it is of interest to maximize callbacks.

2.1. The Halfin–Whitt Limit Regime

We consider a sequence of systems indexed by the arrival rate λ, and let $\lambda\to\infty$. The service rates μ_j, $j\in\mathscr{J}$ are held fixed. The associated total number of agents N^λ increases as λ increases. The routing control in the system with arrival rate λ is $\pi^\lambda := \pi(\lambda, N^\lambda)$. Our convention is to superscript all processes and quantities associated with the system having arrival rate λ by λ.

Assumption 1 (The Halfin–Whitt Limit Regime). (i) *There is heavy traffic; specifically, the number of agents in each pool satisfies*

$$\lim_{\lambda\to\infty}\frac{p_j\mu_j N_j^\lambda}{\lambda} = a_j \quad \textit{for each } j\in\mathscr{J},\ \textit{where } a_j>0 \textit{ and } \sum_{j\in\mathscr{J}} a_j = 1.$$

(ii) *There is square-root safety staffing; specifically,*

$$\lim_{\lambda\to\infty}\frac{\sum_{j\in\mathscr{J}}p_j\mu_j N_j^\lambda-\lambda}{\sqrt{\lambda}} = \beta \quad \textit{for some finite } \beta>0.$$

The Halfin–Whitt regime is thought of as a QED regime because (see Halfin and Whitt 1981 and the extension to the Inverted-V model in Armony 2005):

1. the system load $\rho^\lambda\to 1$ and the limiting fraction of delayed customers is strictly between 0 and 1 (efficiency);
2. the average waiting time of delayed customers is small, of order $1/\sqrt{\lambda}$ (quality).

Note that part (i) of Assumption 1 guarantees that $\rho^\lambda := \lambda/\sum_{j\in\mathscr{J}}p_j\mu_j N_j^\lambda \to 1$ as $\lambda\to\infty$. Furthermore, the limiting fraction of agents in each pool is positive since $N_j^\lambda/N^\lambda \to (a_j/(p_j\mu_j))/(\sum_{j\in\mathscr{J}}a_j/(p_j\mu_j))>0$ as $\lambda\to\infty$, for all $j\in\mathscr{J}$. The condition $\beta>0$ in part (ii) implies the stability condition (2) holds for all λ large enough. The quantity $\beta\sqrt{\lambda}$ is commonly referred to as the safety staff.

Finally, it is useful to define the scaled processes

$$\hat{Q}^\lambda(t) := \frac{Q^\lambda(t)}{\sqrt{\lambda}}, \quad \hat{I}_j^\lambda(t) := \frac{I_j^\lambda(t)}{\sqrt{\lambda}}, \quad j\in\mathscr{J},$$
$$\text{and}\quad \hat{X}^\lambda(t) := \frac{X^\lambda(t)-N^\lambda}{\sqrt{\lambda}}.$$

3. The Approximating DCP

We begin by specifying the DCP that arises under Assumption 1, when formally passing to the limit in the control problem (3), or equivalently, (5), as the arrival rate λ increases to infinity. We define $x^+ := \max(0,x)$, $x^- := -\min(0,x)$ for any $x\in\Re$. Under any nonidling control, $Q^\lambda = (X^\lambda - N^\lambda)^+$, it follows that

$$E[Q^\lambda(\infty)] = \sqrt{\lambda}E[(\hat{X}^\lambda(\infty))^+].$$

Next, we assume that there is a reduction in problem dimensionality, so that in some sense

$$\hat{I}_j^\lambda \approx v_j(\hat{X}^\lambda)(\hat{X}^\lambda)^-, \tag{6}$$

for some function

$$v\colon \Re_- \to \Bigg\{(v_1,\dots,v_J)\colon 0 \le v_j \le 1 \\ \text{for all } j\in\mathcal{J} \text{ and } \sum_{j\in\mathcal{J}} v_j = 1\Bigg\}, \tag{7}$$

where $\Re_- := (-\infty, 0]$. The function v specifies the division of idle agents into pools; for example, if there are $I = (\hat{X}^\lambda)^-$ idle agents, the percentage from pool j is $v_j(\hat{X}^\lambda)$. We do not specify a rigorous weak convergence statement from which (6) follows, although our simulation results in §5 are consistent with this simplification.[1] Then, the problem (5) can be approximated as

$$\mathcal{C}_\star^\lambda \approx \sqrt{\lambda}\Bigg(\operatorname*{minimize}_{\pi^\lambda\in\Pi}\Bigg(cE[\hat{X}^\lambda(\infty;\pi^\lambda)^+] \\ -\sum_{j\in\mathcal{J}}(1-p_j)\mu_j E[v_j(\hat{X}^\lambda(\infty;\pi^\lambda))\hat{X}^\lambda(\infty;\pi^\lambda)^-]\Bigg)\Bigg) \\ +\sum_{j\in\mathcal{J}}(1-p_j)\mu_j N_j, \tag{8}$$

so that specifying the diffusion $\hat{X}$ that approximates $\hat{X}^\lambda$ in the Halfin–Whitt limit regime yields the relevant DCP.

The process $(X^\lambda, I_1^\lambda,\dots,I_J^\lambda)$ is a multidimensional Markov process. Since we assume that callbacks return immediately, X^λ does not change when a callback occurs. Then, X^λ increases by 1 at rate λ and decreases by 1 at a state-dependent rate $\sum_{j\in\mathcal{J}} p_j\mu_j(N_j^\lambda - I_j^\lambda(t))$. The infinitesimal drift is

$$\lim_{h\downarrow 0}\frac{1}{h}E[\hat{X}^\lambda(t+h)-\hat{X}^\lambda(t)\mid \hat{X}^\lambda(t),\hat{I}_1^\lambda(t),\dots,\hat{I}_J^\lambda(t)] \\ = \frac{\lambda-\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\sqrt{\lambda}} + \sum_{j\in\mathcal{J}} p_j\mu_j \hat{I}_j^\lambda(t),$$

which by (6) can by approximated by

$$\frac{\lambda-\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\sqrt{\lambda}} + \sum_{j\in\mathcal{J}} p_j\mu_j v_j(\hat{X}^\lambda(t))\hat{X}^\lambda(t)^-.$$

The infinitesimal variance is

$$\lim_{h\downarrow 0}\frac{1}{h}E[(\hat{X}^\lambda(t+h)-\hat{X}^\lambda(t))^2\mid \hat{X}^\lambda(t),\hat{I}_1^\lambda(t),\dots,\hat{I}_J^\lambda(t)] \\ = \frac{\lambda+\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\lambda} - \frac{\sum_{j\in\mathcal{J}}p_j\mu_j\hat{I}_j^\lambda(t)}{\sqrt{\lambda}},$$

which again by (6) can be approximated by

$$\frac{\lambda+\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\lambda} - \frac{\sum_{j\in\mathcal{J}}p_j\mu_j v_j(\hat{X}^\lambda(t))\hat{X}^\lambda(t)^-}{\sqrt{\lambda}}.$$

Since, from Assumption 1,

$$\frac{\lambda-\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\sqrt{\lambda}} \to -\beta \quad\text{and}\quad \frac{\lambda+\sum_{j\in\mathcal{J}}p_j\mu_j N_j^\lambda}{\lambda}\to 2, \\ \text{as } \lambda\to\infty,$$

we expect that under any control for which (6) holds in an appropriate sense, when λ is large, $\hat{X}^\lambda$ can be approximated by $\hat{X}$ that solves the stochastic equation

$$\hat{X}(t) = \hat{X}(0) + \int_0^t m(\hat{X}(s), v(\hat{X}(s)))\,ds + \sqrt{2}B(t) \tag{9}$$

for

$$m(x,v) = -\beta + \sum_{j\in\mathcal{J}} p_j\mu_j v_j(x)x^-.$$

We write $\hat{X}(\cdot\,; v)$ to denote that $\hat{X}$ solves (9) under the control $v\in\mathcal{V}$. In a slight abuse of notation, we have not specified the extension of the control $v = (v_1,\dots,v_J)$ to the case that $x>0$, because then there is no control decision.

The relevant DCP follows by replacing $\hat{X}^\lambda$ in (8) by $\hat{X}$ and is

$$\mathcal{C}^\star := \operatorname*{minimize}_{v\in\mathcal{V}}\Bigg(cE[\hat{X}(\infty;v)^+] \\ -\sum_{j\in\mathcal{J}}(1-p_j)\mu_j E[v_j(\hat{X}(\infty;v))\hat{X}(\infty;v)^-]\Bigg), \tag{10}$$

where $\mathcal{V}$ is the space of all functions as in (7). We expect that

$$\mathcal{C}_\star^\lambda \approx \sqrt{\lambda}\,\mathcal{C}^\star + \sum_{j\in\mathcal{J}}(1-p_j)\mu_j N_j^\lambda. \tag{11}$$

We denote by $\mathcal{C}(v)$ the expression to be minimized in (10). A control $v^\star\in\mathcal{V}$ is optimal if $\mathcal{C}^\star = \mathcal{C}(v^\star)\le\mathcal{C}(v)$ for all $v\in\mathcal{V}$.

The key to solving the DCP is the following verification theorem.

Theorem 1. *Suppose there exists a twice-continuously differentiable function $V\colon \mathcal{R}\to\mathcal{R}$ and a constant d that solve*

$$\min_{v\in\mathcal{V}}\Bigg\{V''(x) + m(x,v)V'(x) + cx^+ \\ -\sum_{j\in\mathcal{J}}(1-p_j)\mu_j v_j(x)x^-\Bigg\} = d \quad \textit{for all } x\in\mathcal{R}. \tag{12}$$

[1] For example, the approximation (6) is supported under any QIR control defined in Gurvich and Whitt (2009a) by their state-space collapse result Theorem 3.1, when there are no callbacks. We conjecture the approximation also holds when there are callbacks.

Also assume there exist $b_1, b_2 \in \mathcal{R}$ such that $|V(x)| \le b_1x^2 + b_2$ for all $x \in \mathcal{R}$, $E[\hat{X}(0)^2] < \infty$. Then,

$$v^\star(x) = \arg\min_{v \in \mathcal{V}} \left\{ V''(x) + m(x, v)V'(x) + cx^+ - \sum_{j \in \mathcal{J}} (1-p_j)\mu_j v_j(x) x^- \right\} \quad (13)$$

is an optimal control and

$$\mathcal{C}(v^\star) = d;$$

that is, if $\hat{X}$ satisfies (9) *under some admissible control $v \in \mathcal{V}$, then $\mathcal{C}(v) \ge \mathcal{C}(v^\star)$.*

The optimal control $v^\star$ is found by solving (12) and (13). We first solve it when $J = 2$, in §3.1, and then when $J > 2$, in §3.2.

3.1. The DCP Solution When $J = 2$

For the case that $J = 2$, from (13),

$$\begin{aligned}
&\arg\min_{v\in\mathcal{V}} \left\{ V''(x) + m(x,v)V'(x) + cx^+ - \sum_{j=1}^{2} (1-p_j)\mu_j v_j(x) x^- \right\} \\
&= \arg\min_{v\in\mathcal{V}} \left\{ \sum_{j=1}^{2} (p_j\mu_j V'(x) - (1-p_j)\mu_j) v_j(x) x^- \right\} \\
&= \begin{cases} (1,0) & \text{if } p_1\mu_1 V'(x) - (1-p_1)\mu_1 \\ & \qquad \le p_2\mu_2 V'(x) - (1-p_2)\mu_2, \\ (0,1) & \text{otherwise.} \end{cases}
\end{aligned}$$

We conclude that for

$$T(1,2) := \frac{(1-p_2)\mu_2 - (1-p_1)\mu_1}{p_2\mu_2 - p_1\mu_1},$$

an optimal control is

$$v^\star(x) = \begin{cases} (1,0) & \text{if } V'(x) \ge T(1,2), \\ (0,1) & \text{otherwise,} \end{cases} \quad x < 0. \quad (14)$$

Observe that $v^\star$ is a static priority control when either $V'(x) > T(1,2)$ for all $x < 0$, or $V'(x) \le T(1,2)$ for all $x < 0$. Otherwise, $v^\star$ is a state-dependent dynamic control. Recalling the assumption $p_1\mu_1 < p_2\mu_2$, when $p_1 \le p_2$, the pool 2 agents have both a faster effective service rate and a higher resolution probability. Then, intuition suggests that pool 2 agents should never be idled; that is, an optimal control is $v^\star(x) = (1,0)$ for all $x < 0$. To establish this rigorously, it is sufficient to show that the function $V: \mathcal{R} \to \mathcal{R}$ and constant d that solve

$$\begin{aligned} V''(x) - \beta V'(x) + cx = d, \quad & x \ge 0, \\ V''(x) - (\beta + p_1\mu_1 x)V'(x) + (1-p_1)\mu_1 x = d, \quad & x < 0, \end{aligned} \quad (15)$$

also solve (12) and satisfy the conditions of Theorem 1. For this, note that it is straightforward to find analytic expressions for the unique function V and constant d that solve (15) such that V is twice-continuously differentiable. Then, it is also straightforward to show that function V has $V'(x)$ increasing in x and $V'(x) \to (1-p_1)/p_1$ as $x \to -\infty$. Since $p_1 \le p_2$, $(1-p_1)/p_1 \ge T(1,2)$. Therefore, $V'(x) \ge T(1,2)$, which from (14) implies $v^\star(x) = (1,0)$ for all $x < 0$ is an optimal control.

The next question is, what happens when $p_1 > p_2$? Now, there is a trade-off: the pool 1 agents have a higher resolution probability, but the pool 2 agents have a higher effective service rate. Hence, there is no reason to expect that in general a static priority control will be optimal. However, when c is low, so that more importance is placed on call resolution than customer waiting, there is a natural intuition that suggests the static priority control that always idles pool 2 agents is optimal; that is, $v^\star(x) = (0,1)$ for all $x < 0$. Similar methodology as that described in the preceding paragraph suggests finding the unique function V and constant d that solve (15), with p_2 and μ_2 replacing p_1 and μ_1, such that V is twice-continuously differentiable. Then, algebra shows that when

$$c \le C := \frac{\mu_1(p_1 - p_2)}{p_2(p_2\mu_2 - p_1\mu_1)}\beta^2 \cdot \left(1 + \frac{\phi(\beta/\sqrt{p_2\mu_2})}{(\beta/\sqrt{p_2\mu_2})\Phi(\beta/\sqrt{p_2\mu_2})}\right), \quad (16)$$

we have $V'(0) \le T(1,2)$. Since $V'(x)$ increasing in x, we conclude from (14) that $v^\star(x) = (0,1)$ for all $x < 0$ is an optimal control.

Finally, it remains to derive the optimal control $v^\star$ when $p_1 > p_2$ and $c > C$. Intuitively, when x is larger, the system is crowded, and so we want to idle the slower pool. On the other hand, when x is smaller, the system is not crowded, and so we want to idle the faster pool, which has a worse resolution. Based on this intuition, we search for an optimal control within the class of threshold controls, defined as

$$v_L(x) := (\mathbf{1}\{-L \le x < 0\}, \mathbf{1}\{x < -L\}).$$

Now, to use Theorem 1 to establish that v_L is an optimal control, we want to solve for the function $V: \mathcal{R} \to \mathcal{R}$, constant d, and threshold level L in

$$V'(x) = \begin{cases} V_0'(x) & \text{if } x \ge 0, \\ V_1'(x) & \text{if } -L \le x < 0, \\ V_2'(x) & \text{if } x < -L, \end{cases} \quad (17)$$

such that V is twice-continuously differentiable and has $V'(x) < T(1,2)$ when $x < -L$ and $V'(x) \ge T(1,2)$ when $-L \le x < 0$. This follows from Theorem 2 in §3.2 (which shows that a threshold control is optimal for the general J-pool system, possibly with 0 or infinite threshold levels).

In summary, a solution to the DCP when $J=2$ is

$$v^{\star}(x)=\begin{cases}(1,0) & \text{if } p_1\le p_2,\\ (0,1) & \text{if } p_1>p_2 \text{ and } c\le C,\\ (\mathbf{1}\{-L\le x<0\}, & \\ \quad \mathbf{1}\{x<-L\}) & \text{if } p_1>p_2 \text{ and } c>C.\end{cases} \tag{18}$$

We end this subsection by performing a sensitivity analysis on the parameter C defined in (16).

Lemma 1. *When $p_1\mu_1<p_2\mu_2$ and $p_1>p_2$,*

$$\frac{\partial C}{\partial p_1}>0,\quad \frac{\partial C}{\partial \mu_1}>0,\quad \frac{\partial C}{\partial p_2}<0,\quad \frac{\partial C}{\partial \mu_2}<0,\quad \frac{\partial C}{\partial \beta}>0.$$

The first two inequalities in Lemma 1 imply that as long as $p_1\mu_1$ remains below $p_2\mu_2$, increasing p_1 or μ_1 will increase C; that is, the optimal control routes to pool 1 in a larger part of the parameter space. This is not surprising since then either pool 1's resolution probability or its effective service rate has improved. Similarly, the next two inequalities imply that as long as p_2 remains below p_1, increasing p_2 or μ_2 will decrease C; that is, the optimal control routes to pool 2 in a larger part of the parameter space. Again, this is not surprising since then either pool 2's resolution probability or its effective service rate has improved. The intuition for the last inequality is that increasing β increases the safety staff, resulting in reduced queue length, so that the optimal control routes to pool 1 in a larger part of the parameter space.

3.2. The DCP Solution When $J>2$

Recall from the last subsection that when $p_1>p_2$ and the queue-length cost c is large ($c>C$ defined in (16)), a threshold control is optimal. There is a similar optimal solution in the J-pool case, except that a threshold control can have multiple thresholds. The threshold values are used to determine which pool should be idled at time $t>0$, given the value of $\hat{X}(t)$. The number of thresholds is determined by the value of c, and can be any integer value between 0 and $J-1$. In the case that the number of thresholds is zero, a static priority control is optimal. In this subsection, we use Theorem 1 to verify that a threshold control is optimal, and to determine the number of thresholds.

Suppose $V(x)$ and d satisfy the conditions of Theorem 1, and define

$$j^{\star}(x):=\min\left\{\arg\min_{j\in\mathcal{J}}\{V'(x)p_j\mu_j-(1-p_j)\mu_j\}\right\}. \tag{19}$$

Then,

$$\begin{aligned}&\arg\min_{v\in\mathcal{V}}\Big\{V''(x)+m(x,v)V'(x)+cx^{+}\\&\qquad-\sum_{j\in\mathcal{J}}(1-p_j)\mu_j v_j(x^-)x^-\Big\}\\&=\arg\min_{v\in\mathcal{V}}\Big\{\sum_{j\in\mathcal{J}}(p_j\mu_jV'(x)-(1-p_j)\mu_j)v_j(x^-)x^-\Big\}\\&=e_{j^{\star}(x)},\end{aligned}$$

where e_j is the J-dimensional vector that has 0's everywhere except for a 1 in the jth position. It follows from (13) that $v^{\star}(x)=e_{j^{\star}(x)}$ is an optimal control.

We begin by arguing intuitively that there exists a pool set $\mathcal{K}^{\star}\subseteq\mathcal{J}$ having K pools such that the pools in the set $\mathcal{J}-\mathcal{K}^{\star}$ are never idled ($j^{\star}(x)\notin\mathcal{J}-\mathcal{K}^{\star}$ for any $x<0$). We will later verify rigorously that this assertion is correct. First, observe that if there exist two pools i and j such that $j>i$ and $p_j\ge p_i$, then pool j has both the higher effective service rate and the higher resolution probability, and so should never be idled. In other words, the pools in the set $\mathcal{K}^{\star}$ can be relabeled so that

$$p_1\mu_1<p_2\mu_2<\cdots<p_K\mu_K \quad\text{and}\quad p_1>p_2>\cdots>p_K.$$

Then, for any two pools in the set $\mathcal{K}^{\star}$, there is a trade-off between resolution probability and effective service rate. Next, consistent with the definition of $T(1,2)$ in §3.1, let

$$T(i,j):=\frac{(1-p_j)\mu_j-(1-p_i)\mu_i}{p_j\mu_j-p_i\mu_i},\quad i<j\in\mathcal{J} \tag{20}$$

measure the ratio of the difference between the call back rates and the resolution rates of pools i and j. For $i<j<k\in\mathcal{J}$, $T(i,j)<T(j,k)$ suggests that pool j combines the strengths of pools i and k; that is, its resolution is not too much worse than that of pool i (implying the numerator of $T(i,j)$ is small) and its effective service rate is not too much worse than that of pool k (implying the denominator of $T(j,k)$ is small). Hence, pool j should never be idled. This can also be seen algebraically by first noting that $T(i,j)\le T(i,k)\le T(j,k)$ (see the proof of Corollary 1 in the EC), and then observing that

- when $V'(x)\ge T(i,k)$, also $V'(x)\ge T(i,j)$, which is equivalent to

$$p_j\mu_jV'(x)-(1-p_j)\mu_j\ge p_i\mu_iV'(x)-(1-p_i)\mu_i;$$

- when $V'(x)<T(i,k)$, also $V'(x)<T(j,k)$, which is equivalent to

$$p_j\mu_jV'(x)-(1-p_j)\mu_j>p_k\mu_kV'(x)-(1-p_k)\mu_k.$$

The definition of $j^{\star}$ in (19) then implies that $j^{\star}(x)\neq j$ for all $x\le 0$. We conclude that the pools in the set $\mathcal{K}^{\star}$ should be able to be relabeled so that

$$\begin{gathered}p_1\mu_1<p_2\mu_2<\cdots<p_K\mu_K,\\ p_1>p_2>\cdots>p_K,\\ T(1,2)>T(2,3)>\cdots>T(K-1,K).\end{gathered} \tag{21}$$

Our next theorem shows that for a system with pool set $\mathscr{K}^\star$ having parameters that satisfy (21), a threshold control is optimal, and specifies the number of strictly positive threshold levels.

Definition 1 (Threshold Control). For a system with K pools, given $K-1$ thresholds in a vector $\vec{L} = (L_1, L_2, \dots, L_{K-1})$ having $0 \le L_1 \le L_2 \le \cdots \le L_{K-1}$, a threshold control is defined as

$$v_{\vec{L}}(x) = (\mathbf{1}\{-L_1 \le x < 0\}, \mathbf{1}\{-L_2 \le x < -L_1\}, \dots, \\ \mathbf{1}\{-L_{K-1} \le x < -L_{K-2}\}, \mathbf{1}\{x < -L_{K-1}\}).$$

Theorem 2. *For each $c > 0$, there exists an increasing function $V'(x)$ and a constant d that satisfy the conditions of Theorem 1 with $\mathscr{V} := \mathscr{V}(\mathscr{K}^\star)$ defined as in (7), except using the set $\mathscr{K}^\star$ instead of $\mathscr{J}$. Furthermore, there exists a sequence of finite constants*

$$C_0 := 0 < C_1 < C_2 < \cdots < C_{K-1} < C_K := \infty$$

such that if $c \in (C_{k-1}, C_k]$ for some $k \in \mathscr{K}^\star$, then the threshold control $v_{\vec{L}}$ with $K-k$ nonzero thresholds is optimal,

$$v^\star = v_{\vec{L}},$$

where

$$L_i = \begin{cases} -V'^{-1}(T(i, i+1)) & i = k, k+1, \dots, K-1, \\ 0 & i = 1, 2, \dots, k-1, \end{cases}$$

and $0 < L_k < \cdots < L_{K-1}$. The parameters $C_1, \dots, C_{K-1}$ and $L_1, \dots, L_{K-1}$ can be found by a sequence of one-dimensional searches.

It is intuitive that as c increases, the threshold levels should increase, because it implies that the pools with the lower effective service rate will idle more. Furthermore, as c becomes very large, the optimal control will be a static priority control that only idles the pool with the smallest effective service rate.

Lemma 2. *When $c > C_1$, L_i is increasing in c, for all $i \in \{1, 2, \dots, K-1\}$. When $c \to \infty$, $L_i \to \infty$.*

Note that Lemma 2 is only relevant for $c > C_1$ because otherwise all threshold levels are zero.

The optimal control for the J-pool system follows from the optimal control for the system with pool set $\mathscr{K}^*$. This can be seen from the following two corollaries to Theorem 2.

Corollary 1. *Assume the pools satisfy (1), and for some $i < j < k \in \mathscr{J}$, $T(i, j) \le T(j, k)$. If a function V and a constant d satisfy the conditions of Theorem 1 and $v^{\mathscr{J}'}$ is defined by (13) with $\mathscr{V}(\mathscr{J}')$ for $\mathscr{J}' := \mathscr{J} - \{j\}$ replacing $\mathscr{V}$ defined in (7), then an optimal control (when $\mathscr{V}(\mathscr{J}')$ does not replace $\mathscr{V}$) is*

$$v^\star = (v_1^{\mathscr{J}'}, \dots, v_{j-1}^{\mathscr{J}'}, 0, v_{j+1}^{\mathscr{J}'}, \dots, v_{J-1}^{\mathscr{J}'}).$$

Corollary 2. *Assume the pools satisfy (1), and that there exists some $i < j < \mathscr{J}$ such that $p_j \ge p_i$. If a function V and a constant d satisfy the conditions of Theorem 1 and $v^{\mathscr{J}'}$ is as defined in (13) with $\mathscr{V}(\mathscr{J}')$ for $\mathscr{J}' := \mathscr{J} - \{j\}$ replacing $\mathscr{V}$ defined in (7), then an optimal control (when $\mathscr{V}(\mathscr{J}')$ does not replace $\mathscr{V}$) is*

$$v^\star = (v_1^{\mathscr{J}'}, \dots, v_{j-1}^{\mathscr{J}'}, 0, v_{j+1}^{\mathscr{J}'}, \dots, v_{J-1}^{\mathscr{J}'}).$$

Finally, Corollaries 1 and 2 can be used iteratively to determine a set $\mathscr{K}^\star \subset \mathscr{J}$ such that the pools in the set $\mathscr{J} - \mathscr{K}^\star$ are never idled. Furthermore, the pools satisfy (21) under an appropriate relabeling. In summary, the optimal control for the J-pool system follows from $v^\star$ given in Theorem 2 for the system with pool set $\mathscr{K}^*$.

An optimal control for the J-pool system is written as follows. Let τ: $\mathscr{J} \to \mathscr{K}^\star$ be a mapping such that $\tau(j) = k$ if $j \in \mathscr{K}^\star$ and pool j is the kth pool under the relabeling that satisfies (21), and $\tau(j) = 0$ if $j \notin \mathscr{K}^\star$. Let $v^{\mathscr{K}^\star}$ be the control specified in Theorem 2 over the control set $\mathscr{V}(\mathscr{K}^\star)$. Define $v^\star$ to be the vector having components

$$v_j^\star = \begin{cases} v_{\tau(j)}^{\mathscr{K}^\star} & \text{if } \tau(j) \ne 0, \\ 0 & \text{otherwise}, \end{cases} \quad j \in \mathscr{J}.$$

Theorem 3. *The control $v^\star$ is an optimal control for the J-pool system.*

In general, $v^\star$ has a threshold structure. Note that if $p_1 = \min_{j \in \mathscr{J}}\{p_j\}$, then $\mathscr{K}^\star = \{1\}$, so that $v^\star = e_1$. Pool 1 is always idled because it has the lowest resolution probability and the lowest effective service rate; hence, almost all the idle agents are in Pool 1. Also, when c is small enough ($c < C_{K-1}$), $v^\star$ always idles the pool with the relabeled index K in (21). This is because the cost of queue length is small and the relabeled pool K has the lowest resolution probability of the pools in $\mathscr{K}^\star$. Finally, since our initial assumption only assumed pools were ordered according to the values of their effective service rates, so that $p_1\mu_1 < \cdots < p_J\mu_J$, Theorem 3 gives an optimal control for the entire parameter space (β, μ_j, and p_j, $j \in \mathscr{J}$) of the DCP for the J-pool system.

Remark 1. Recall that in §2 we showed that any routing control $\pi^\star \in \Pi$ that achieves the minimum in the original objective (3) is on the efficient frontier with respect to the steady-state average waiting time and call resolution. Similarly, any control $v \in \mathscr{V}$ that achieves the minimum in (10) is on the efficient frontier. If $l_1, l_2, \dots, l_{K-1}$ defined through Equations (6)–(13) in the EC are continuous in c, then it follows that $E[\hat{X}(\infty; v^\star)^+]$ is continuously decreasing in c, and $\sum_{j \in \mathscr{J}}(1-p_j)\mu_j E[v_j(\hat{X}(\infty; v^\star))\hat{X}(\infty; v^\star)^-]$ is continuously increasing in c. Noting that the p-rule solves (8) when $c = 0$ and the $p\mu$-rule solves (8) as c becomes arbitrarily large, it follows that the set of points found by varying c from 0 to ∞ in (8) generates the entire efficient frontier.

4. The Proposed RPT Control

In this section, we translate the optimal control $v^\star$ that solves the approximating DCP in §3 into a routing control for the original system. More specifically, when the total number of agents at time $t > 0$, $\hat{I}(t)$, is positive, the control $v^\star$ determines the pool $j^\star(t)$ that should have the lowest routing priority (which is the pool corresponding to the component of $v^\star(\hat{X}^\lambda(t))$ that is 1). Because the control $v^\star$ is a threshold control on the reduced pool set $\mathcal{K}^\star \subseteq \mathcal{J}$, we name our proposed control the RPT control. We expect that at any time $t > 0$, almost all idle agents are from the pool $j^\star(t)$, and all other pools have very few idle agents.

The structure of the RPT control can be stated without reference to the DCP or its solution $v^\star$, and this is our objective in this section. In particular, for any given set of parameter values, we provide the number of potentially nonzero threshold levels, which is between 0 and $J-1$. The threshold values can be set either as the values in Theorem 2 (which gives the reduced pool DCP solution), multiplied by $\sqrt{\lambda}$, or they can be found by numeric search. The use of Theorem 2 also evidences the number of strictly positive threshold values, according to the value of c.

We state the structure of the RPT control first in the case that $J=2$ (§4.1), then in the case that $J=3$ (§4.2), and finally for any J (§4.3). For intuition, it is helpful to recall that the effective service rates are ordered from lowest to highest, as in (1).

4.1. The RPT Control When $J=2$

The RPT control is either the p-rule, the $p\mu$-rule, or a threshold control, depending on the system parameters. The *threshold control* $\pi_{\mathrm{T}}(L)$ for $L > 0$ assigns a newly arriving customer at time $t > 0$ to an idle agent according to the priority rule ($i \succ j$ denotes pool i has higher priority than pool j in routing):

$$\pi_{\mathrm{T}}(L) := \begin{cases} 1 \succ 2 & \text{if } I(t) > L, \\ 2 \succ 1 & \text{if } I(t) \le L; \end{cases}$$

otherwise, in the case that no agent is free, the customer queues. Table 1 provides the RPT control for a two-pool system, where the value of C is given in (16) in §3.1. Note that when c is small, so that minimizing queue length is much less important than minimizing callbacks, the RPT control is the p-rule. For larger c, the RPT control uses a threshold L to trade off service rate and call resolution. It follows from Lemma 2 that as c goes to ∞, L tends to ∞, so that the RPT control will become equivalent to the $p\mu$-rule.

Table 1 **Two-Pool RPT Control**

Parameter values	Pool priority order
$p_1 \le p_2$	$2 \succ 1$ ($p\mu$-rule equals p-rule)
$p_1 > p_2$ and $c \le C$	$1 \succ 2$ (p-rule)
$p_1 > p_2$ and $c > C$	$\pi_{\mathrm{T}}(L)$

4.2. The RPT Control When $J=3$

It is helpful to let $\pi_{\mathrm{RPT}}(j_1, j_2)$, for $j_1, j_2 \in \mathcal{J}$, denote the two-pool RPT control having pool priority order defined in Table 1, where we pretend the system consists of only the pools j_1 and j_2, with the number of idle agents $I(t)$ redefined as $I_{j_1}(t) + I_{j_2}(t)$. Next, the *two threshold control* $\pi_{\mathrm{T}}(L_1, L_2)$ for $L_2 > L_1 > 0$ and $M := (L_1 + L_2)/2$ assigns a newly arriving customer at time $t > 0$ to an idle agent according to the priority order given in the following table:

System state	Pool priority order
$0 < I(t) \le L_1$	$3 \succ 2 \succ 1$ ($p\mu$-rule)
$L_1 < I(t) \le M$	$3 \succ 1 \succ 2$
$M < I(t) \le L_2$	$1 \succ 3 \succ 2$
$I(t) > L_2$	$1 \succ 2 \succ 3$ (p-rule)

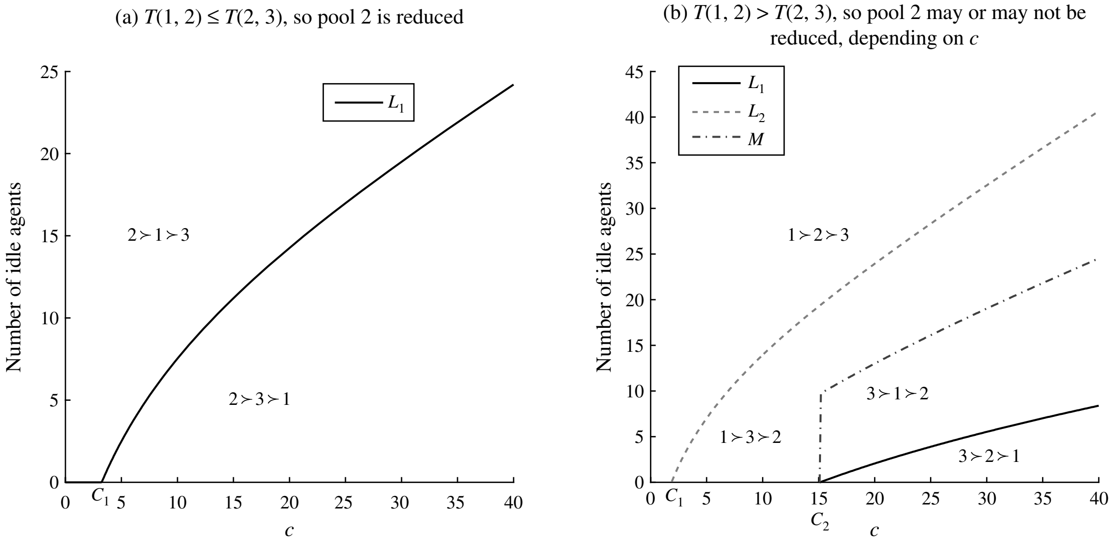
Otherwise, in the case that no agent is free, the customer queues. In particular, the pool priority order places more emphasis on resolution probabilities as the number of idle agents increases. The value of M is chosen so that the RPT control can be thought of as giving priority in accordance with the $p\mu$-rule excluding pool 2 when $I(t)$ is lower and the p-rule excluding pool 2 when $I(t)$ is larger. Pool 2 is excluded because it is the pool that should be idled when we follow the DCP solution.

Table 2 provides the RPT control for a three-pool system, where the values of C_1 and C_2 are given in Theorem 2 in §3.2. Although C_1 does not explicitly appear in the table, it is necessary to know C_1 to

Table 2 **Three-Pool RPT Control**

Parameters	Pool priority order	Intuitive explanation
$p_3 \ge \min\{p_1, p_2\}$	$3 \succ \pi_{\mathrm{RPT}}(1,2)$	Pool 3 dominates at least one pool in both metrics; it should never be idled
$p_2 \ge p_1 > p_3$	$2 \succ \pi_{\mathrm{RPT}}(1,3)$	Pool 2 dominates pool 1 in both metrics; it should never be idled
$p_1 > p_2 > p_3$ and $T(1,2) \le T(2,3)$	$2 \succ \pi_{\mathrm{RPT}}(1,3)$	$T(1,2) \le T(2,3)$ suggests that pool 2 has a resolution not too much worse than pool 1 and an effective service rate not too much slower than pool 3; it should never be idled
$p_1 > p_2 > p_3$ and $T(1,2) > T(2,3)$ and $c \le C_2$	$1 \succ \pi_{\mathrm{RPT}}(2,3)$	The cost of queue length is small, pool 1 has the best resolution should never be idled
$p_1 > p_2 > p_3$ and $T(1,2) > T(2,3)$ and $c > C_2$	$\pi_{\mathrm{T}}(L_1, L_2)$	The cost of queue length is high enough that no pool will ever always have priority

Figure 2 **RPT Control When $p_1 > p_2 > p_3$ in a Three-Pool System**


(a) $T(1, 2) \le T(2, 3)$, so pool 2 is reduced

(b) $T(1, 2) > T(2, 3)$, so pool 2 may or may not be reduced, depending on c

Notes. In (a), $\vec{\mu} = (3, 6, 15)$, $\vec{p} = (0.99, 0.8, 0.5)$, $\vec{N} = (25, 25, 25)$, and $\rho = 0.9$. In (b), we change p_2 to 0.6 and keep all other parameter values unchanged.

use $\pi_{\text{RPT}}(j_1, j_2)$ to interpret the pool priority order. Also, when reading Table 2, it is helpful to recall that $T(i, j), i < j \in \mathcal{J}$ defined in (20) measures the ratio of the difference between the callback rates and resolution rates of pools i and j.

Observe that in four of five cases, the RPT control gives one pool the highest priority, meaning agents in that pool are not idled. We consider that pool to be "reduced." Then, the remaining pools are ordered according to the RPT control defined for two pools. In the last case, no pool can be reduced.

Figure 2 provides illustrations of the RPT control when $p_1\mu_1 < p_2\mu_2 < p_3\mu_3$ and $p_1 > p_2 > p_3$, for both the subcases that $T(1, 2) \le T(2, 3)$ and $T(1, 2) > T(2, 3)$. In Figure 2(a), pool 2 is reduced, so that the priority order of pools 1 and 3 is in accordance with the two-pool RPT control in Table 1. Notice that when $c \le C_1$,[2] pools 1 and 3 are ordered as in the p-rule. Otherwise, their priority order is determined by a threshold (L_1) on the number of idle agents. In Figure 2(b), pool 2 is not reduced. Then, when $c \le C_1$, the priority order of the pools is as the p-rule. When $c \in (C_1, C_2]$, pool 1 is never idled, and the priority order of pool 2 and pool 3 is determined by a single threshold (L_2). Finally, when $c > C_2$, there are two thresholds (L_1 and L_2) used to determine the priority of the three pools. Furthermore, when pool 2 has the lowest priority, we use a middle threshold M (defined latter in Definition 2) to determine the priority order of pools 1 and 3 (according to either the $p\mu$-rule excluding pool 2 when the number of idle agents is lower or the p-rule excluding pool 2 when the number of the idle agents is larger).

Remark 2 (Why Is the RPT Control in the Last Case of Table 2 Not Defined Recursively?). The reader may wonder if the threshold levels L_1 and L_2 can be determined separately, using the appropriate thresholds from $\pi_{\text{RPT}}(1, 2)$ and $\pi_{\text{RPT}}(2, 3)$. This does not work, because when we choose L_1 and L_2, we must account for "joint" pool performances. In other words, when there are two thresholds involved, it is not possible to "reduce" a pool and rely on the two-pool RPT control.

Remark 3 (Why Does the Pool Reduction Require Conditions on $T(i, j)$?). We explain the intuition for why pool 2 is reduced when $T(1, 2) \le T(2, 3)$ graphically, using Figure 3. First, for each pool j, we draw a point $(\mu_j, p_j\mu_j)$ on the $\mu - p\mu$ plane. Next we draw the line segment connecting $(\mu_1, p_1\mu_1)$ and $(\mu_3, p_3\mu_3)$, defined by

$$p\mu = \frac{p_3\mu_3 - p_1\mu_1}{\mu_3 - \mu_1}\mu + \frac{\mu_1\mu_3(p_1 - p_3)}{\mu_3 - \mu_1}, \quad \mu \in [\mu_1, \mu_3]. \tag{22}$$

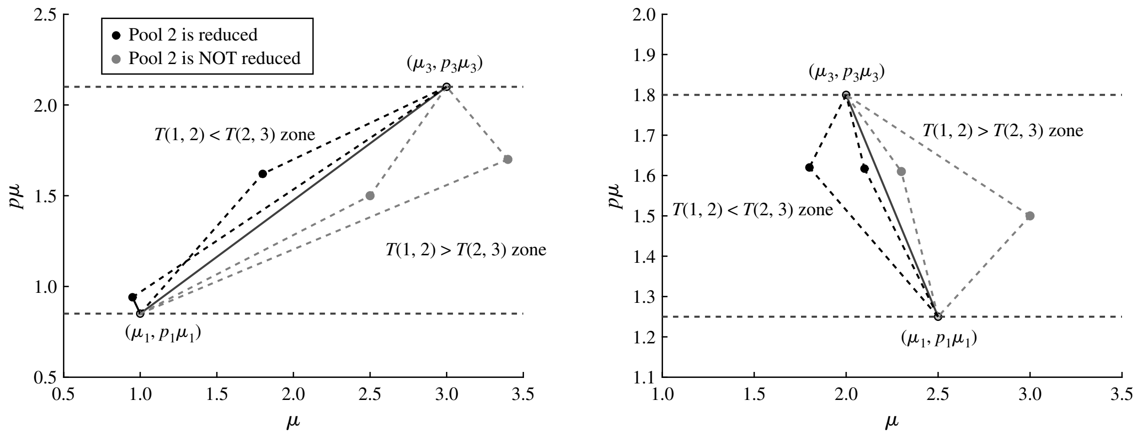
Claim (Verified at the end of this remark). The condition $T(1, 2) < T(2, 3)$ is equivalent to the point $(\mu_2, p_2\mu_2)$ lying on the left of the line (22) shown in Figure 3.

Any point $(\mu, p\mu)$ on the line segment (22) represents a virtual pool of agents that can be viewed as a combination of agents from pools 1 and 3. For example, if $\mu = \alpha\mu_1 + (1 - \alpha)\mu_3$ for some $\alpha \in (0, 1)$, then, from (22), $p\mu = \alpha p_1\mu_1 + (1 - \alpha)p_3\mu_3$. Idling one agent from the virtual $(\mu, p\mu)$ pool is equivalent to idling α pool 1 agents and $(1 - \alpha)$ pool 3 agents.

Suppose $(\mu_2, p_2\mu_2)$ locates on the left of the line segment, and consider the choice between idling one pool 2 agent and one virtual pool agent. If there is

[2] Note that when pool 2 is reduced as in Figure 2(a), C_1 is defined as C in (16), except that p_2 and μ_2 are replaced by p_3 and μ_3.

Figure 3 **Reduction of Pool *j* Depends on Its Location**



a virtual pool that has worse performance, then we will idle an agent from that virtual pool. Define a virtual pool $(\mu_m, p_m\mu_m)$ by $\mu_m = \alpha\mu_1 + (1-\alpha)\mu_3$ using α such that $p_2\mu_2 = \alpha p_1\mu_1 + (1-\alpha)p_3\mu_3$. Then, $p_m\mu_m = p_2\mu_2$. Furthermore, as can be seen from Figure 3, $\mu_2 < \mu_m$. Hence, $p_2 > p_m$. In summary, the virtual pool has worse performance because it has the same effective service rate and lower resolution probability. We prefer to idle the virtual pool agent. The pool 2 agent is not idled, and so should have routing priority, meaning pool 2 routing priorities need not be assigned dynamically. Pool 2 is reduced.

Verification of the Claim. First note that $T(1,2) = ((1-p_2)\mu_2 - (1-p_1)\mu_1)/(p_2\mu_2 - p_1\mu_1) < T(2,3) = ((1-p_3)\mu_3 - (1-p_2)\mu_2)/(p_3\mu_3 - p_2\mu_2)$ is equivalent to

$$\frac{\mu_2-\mu_1}{p_2\mu_2-p_1\mu_1} < \frac{\mu_3-\mu_2}{p_3\mu_3-p_2\mu_2}. \qquad (23)$$

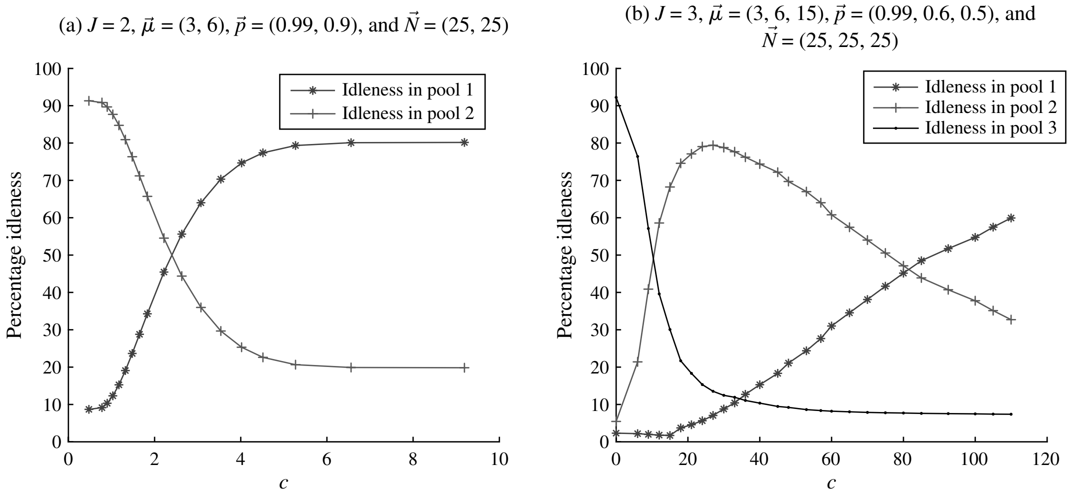
Consider an example of pool 2 lying at the left of the line segment connecting pool 1 and pool 3. If $\mu_2 \le \min\{\mu_1, \mu_3\}$, then the left-hand side of (23) is nonpositive, whereas the right-hand size is positive, and (23) is valid. If $\mu_1 < \mu_2 < \mu_3$ or $\mu_1 > \mu_2 > \mu_3$, then (23) is equivalent to

$$\frac{p_2\mu_2-p_1\mu_1}{\mu_2-\mu_1} > \frac{p_3\mu_3-p_2\mu_2}{\mu_3-\mu_2},$$

meaning that the slope of the line segment connecting pool 1 and pool 2 is larger than the slope of the line segment connecting pool 2 and pool 3, this can be seen from Figure 4.

Remark 4 (Idleness Allocation). When the RPT control has a threshold structure (with nonzero and noninfinite thresholds), the steady-state idleness allocation in all the pools is strictly between 0 and 1; see Figure 4. Then, the RPT control exactly coincides with the (limiting) threshold control shown in Armony and Ward (2010) to minimize average waiting time subject to a steady-state fairness constraint (when the μ in Armony and Ward 2010 is replaced by $p\mu$). This shows that considering the dual objective of minimizing queue length (equivalently customer waiting) and callback rate can lead naturally to a fair control,

Figure 4 **Idleness Allocation in Each Pool**


(a) $J = 2$, $\vec{\mu} = (3, 6)$, $\vec{p} = (0.99, 0.9)$, and $\vec{N} = (25, 25)$

(b) $J = 3$, $\vec{\mu} = (3, 6, 15)$, $\vec{p} = (0.99, 0.6, 0.5)$, and $\vec{N} = (25, 25, 25)$

provided the parameters satisfy appropriate conditions (in particular, the last row of Table 1 when $J=2$ and the last row of Table 2 when $J=3$).

4.3. The RPT Control with General J

The RPT control when $J=3$ can be naturally extended to $J>3$. We observe in the case of $J=3$ that in four of five divisions of the parameter space, there is one pool that always has the highest priority in routing. In the general J pool case, the highest priority may be given to a set of pools $\mathcal{J}-\mathcal{K}^\star$, where $\mathcal{K}^\star$ is defined as in §3.2. Specifically, $\mathcal{K}^\star \subseteq \mathcal{J}$ is the set of $K \le J$ pools whose indices can be relabeled so that in addition to the assumed condition

$$p_1\mu_1 < p_2\mu_2 < \cdots < p_K\mu_K,$$

it is also true that

$$T(1,2) > T(2,3) > \cdots > T(K-1,K), \tag{24}$$

$$p_1 > p_2 > \cdots > p_K\mu_K. \tag{25}$$

The priority order of the pools in the set $\mathcal{K}^\star$ is determined by a threshold control with threshold levels $L_0 := 0 \le L_1 \le L_2 \le \cdots \le L_{K-1} < L_K := \infty$. The number of strictly positive threshold levels depends on the value of c, and exactly corresponds to the number of positive threshold levels in Theorem 2. (Note that the issue of positive threshold levels did not arise in §§4.1 and 4.2, because we explicitly separated out these cases in Tables 1 and 2.)

Definition 2 (The RPT Control). Upon the arrival of a customer at time $t>0$, if no agent is idle, the customer queues; otherwise, the customer will be routed to an available agent in accordance with the following (potentially state-dependent) priority order for the pools:

- The pools in the set $\mathcal{J}-\mathcal{K}^\star$ have the highest priority, equally.
- Pool $j^\star$ that satisfies $L_{j^\star-1} < I(t) \le L_{j^\star}$ has the lowest priority.
- The order of the remaining pools is determined by the $p\mu$-rule if $I(t) \le M$, and is otherwise determined by the p-rule, where

$$M := \begin{cases} \dfrac{\sum_{j=1}^{K-1} L_j}{\sum_{j=1}^{K-1} \mathbf{1}\{L_j>0\}} & \text{if } \displaystyle\sum_{j=1}^{K-1} \mathbf{1}\{L_j>0\} > 1, \\ 0 & \text{otherwise.} \end{cases}$$

The value M is set to be in the middle of the positive thresholds values. This is consistent with prioritizing the pools with higher resolution probability when there are many idle agents, and prioritizing the pools with higher effective service rate when there are few idle agents. Note that the value of M is set in accordance with our judgment, because the DCP solution gives no guidance on the priority ordering of the remaining pools in the third bullet point.

Figure 5 RPT Control of a Five-Pool System

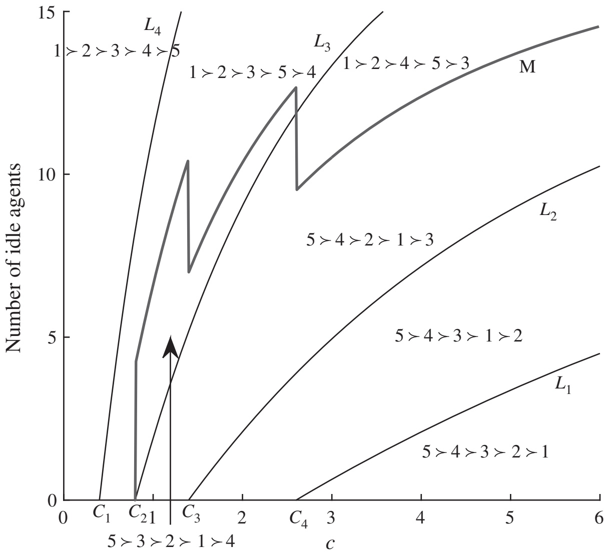


Figure 5 shows the RPT control when $J=5$ and the conditions (24) and (25) are satisfied (in addition to the assumed upfront ordering of the pools from lowest to highest effective service rate in (1)). We see that when $c \le C_1$, the RPT control has no threshold and degenerates to p-rule. As c becomes larger, the RPT control has more and more thresholds and more and more pools may sometimes be idled. When $c > C_4$, the RPT control has four thresholds, meaning that each pool is sometimes idled. When the system state is between the thresholds L_j and L_{j+1}, pool j is idled, and the remaining pools are prioritized in accordance with the p-rule when the number of idle agents exceeds M, and are otherwise prioritized in accordance with the $p\mu$-rule. As $c \to \infty$, all the thresholds go to ∞, and the RPT controls behaves like the $p\mu$-rule.

In general, the RPT control is a dynamic control that uses thresholds to determine pool priorities. The RPT control becomes a static control that always assigns the lowest priority to the pool with the largest index in $\mathcal{K}^\star$ when $\mathcal{K}^\star$ has cardinality $K>1$ and $c \le C_1$, or $K=1$.

5. The Efficient Frontier

We expect that the RPT control is on the efficient frontier as λ becomes large. This is because the RPT control is motivated by the control $v^\star$ in §3 that solves the diffusion control problem that arises as an approximation to (3) as λ becomes large. Our objective in this section is to use simulation to show that the RPT control is indeed on the efficient frontier. This requires comparison controls.

The $p\mu$-rule and the p-rule are natural comparison controls. However, the $p\mu$-rule results in most all of the idle agents being from pool 1, and the p-rule results in most all the idle agents being from the pool with the smallest p_j, $j \in \mathcal{J}$. Therefore, we also consider controls in the literature that can produce an entire range of idleness allocations for each pool, anywhere from 0 to 1.

One such control is the family of queue-and-idleness ratio (QIR) routing controls introduced in Gurvich and Whitt (2009a). We consider a simple QIR control for the inverted-V model that is specified in terms of static idleness ratios $f_j \in [0,1]$, $j \in \mathcal{J}$, having $\sum_{j\in\mathcal{J}} f_j = 1$. Then, upon the arrival of a customer at time $t > 0$, the QIR control routes the customer to an available agent in pool

$$j^\star := j^\star(t) \in \underset{j\in\mathcal{J},\, I_j(t)>0}{\arg\max}\{I_j(t) - f_j I(t)\},$$

and if there are no agents available, the customer queues. Note that when λ is large, consistent with the results of Gurvich and Whitt (2009a) when there are no callbacks, our simulations show that

$$I_j(t) \approx f_j I(t), \quad \text{for all } j \in \mathcal{J}.$$

This confirms our expectation that the reduction in problem dimensionality assumed in the informal derivation of the DCP in the first paragraph of §3 holds.

Another natural comparison control is the longest-weighted-idleness control in Definition 3 in Ward and Armony (2013), which extends the longest-idle-server-first control in Atar (2008). This control routes an arriving customer to the agent that has the longest weighted idle time. Theorem 5 in Ward and Armony (2013) shows that this control is asymptotically equivalent to QIR.

A final natural comparison control is the randomized-most-idle (RMI) routing control in Mandelbaum et al. (2012). This control routes an arrival to one of the pools with available agents, with probability that equals the fraction of idle agents. The RMI control is asymptotically equivalent to a QIR control; see Corollary 1 in Mandelbaum et al. (2012).

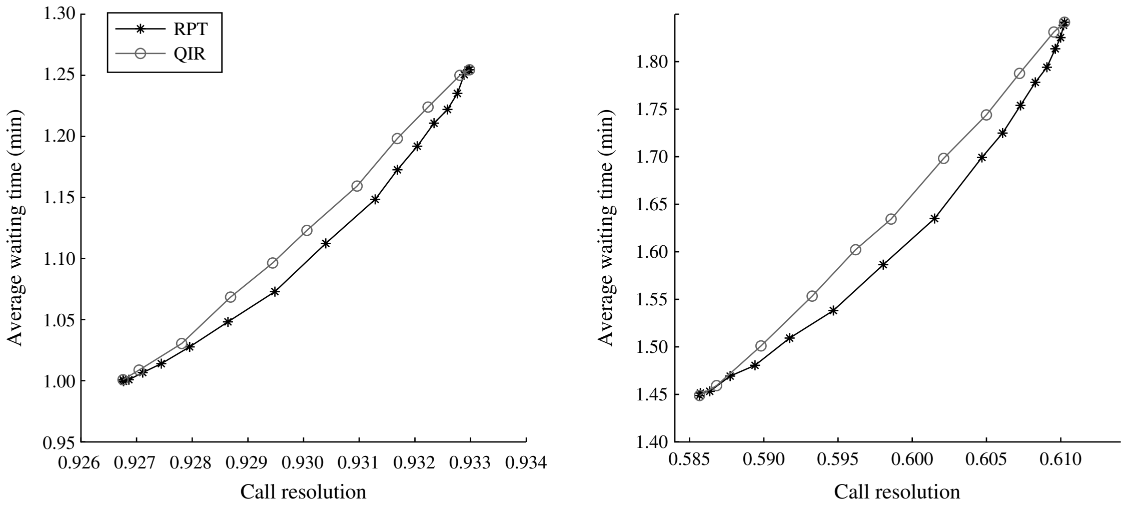
It follows that to show the RPT control is on the efficient frontier, it is reasonable to begin by simulating only the RPT and QIR controls. We choose the simulation parameters based on the empirical data used by Mehrotra et al. (2012). In their electronic companion Appendix B, they describe a call center data set in which there are four call types and 228 agents, with each agent capable of handling all four call types. The agents are clustered into 20 pools according to their service speed and resolution probability for each call type. Each pool has 1 to 39 homogeneous agents. To be consistent with our model, we focus on one call type. We simulate the inverted-V model with two and three pools, and our choices for the service speeds and resolution probabilities are guided by the empirical observations in Table EC-1 in the EC. Specifically, when $J = 2$, we let $(\vec{\mu}, \vec{p}) = ((3,6),(0.99,0.90))$, $((3,12),(0.99,0.50))$, and when $J = 3$, we let $(\vec{\mu}, \vec{p}) = ((3,6,15),(0.99,0.80,0.50))$, $((3,6,15),(0.99,0.60,0.50))$. We fix the size of each pool to be 25, in the same order with the data in Mehrotra et al. (2012). (See the EC for details on the simulation study setup, and summary details on the aforementioned data.)

All of the figures reporting our simulation results (Figures 6–8) display the steady-state call resolution $p(\infty;\pi)$ on the x-axis and the steady-state average waiting time $E[W_\Sigma(\infty;\pi)]$ in minutes on the y-axis. Then, the figures capture the resulting decrease in call resolution of the system would experience for any given decrease in average waiting time.

We begin by reporting our simulation results of a two-pool inverted-V model in Figure 6. For this, we vary c from 0 to ∞ to produce a family of RPT controls with threshold levels that vary so that each pool experiences idle time allocation between 0 and 1. Second, for the QIR control, we let $f_1 \in (0.0, 0.1, \ldots, 1.0)$ and $f_2 = 1 - f_1$, so that again the idle time allocation experienced by each pool varies between 0 and 1. We plot the simulated performance for each control from both families of RPT controls and QIR controls. Note the point that corresponds to the highest call resolution and highest average waiting time shows the performance of the p-rule (RPT control with threshold 0, or QIR control with $f_1 = 0$), and the point that corresponds to the lowest call resolution and lowest average waiting time shows the performance of the $p\mu$-rule (RPT control with threshold ∞, or QIR control with $f_1 = 1$). We observe that in Figure 6, both (a) and (b), the RPT control is on the efficient frontier, and the QIR control is not. This is not surprising given that the diffusion approximation in (9) associated with the RPT control has $v^\star(x) = (\mathbf{1}\{-L \le x < 0\}, \mathbf{1}\{x < -L\})$, and the one associated with the QIR control has $v(x) = (f_1, 1 - f_1)$. (We provide additional simulation results regarding the effect of larger and smaller values of ρ on the efficient frontier in the EC.)

The efficient frontier in Figure 6 (and in all the figures in this section) is small in the sense that the average waiting times are all within half a minute, and the call resolutions are all within 3%. This is consistent with the difference in average speed of answer and call resolution shown in Figure 1 in Mehrotra et al. (2012) when we compare the performance of the p-rule and the $p\mu$-rule. (The remaining routing rules shown there do not apply to the inverted-V model.) Still, the trade-off decision between emphasizing waiting time and call resolution is important because (1) small improvements in the experience of each individual customer can lead to large improvements in the performance of the entire system, and

Figure 6 Simulated Comparison Between Change to RPT and QIR Controls in a Two-Pool System



(a) $\vec{\mu} = (3, 6)$, $\vec{p} = (0.99, 0.90)$, $\vec{N} = (25, 25)$, and $\rho = 0.9$

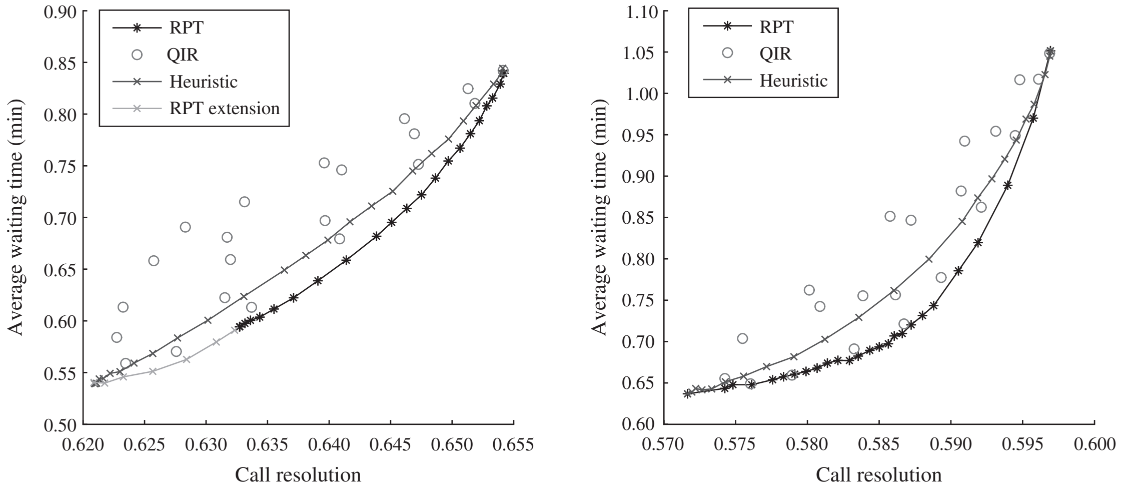
(b) $\vec{\mu} = (3, 12)$, $\vec{p} = (0.99, 0.50)$, $\vec{N} = (25, 25)$, and $\rho = 0.9$

(2) developing the efficient frontier for the inverted-V model analytically in the QED regime can be a stepping stone to developing the efficient frontier in the more general model with heterogeneous customers and heterogeneous agents.

Recalling that the simple p-rule is focused on maximizing the call resolution and the simple $p\mu$-rule is focused on minimizing the average waiting time suggests the following routing heuristic: when there are many idle agents ($I(t) > M$), route customers in accordance with the p-rule, but when there are few idle agents ($I(t) \leq M$), route customers in accordance with the $p\mu$-rule. Then, by varying the value of M from 0 to ∞, we can find all combinations of call resolution and average waiting time that are achievable by this heuristic routing control. Note that in a two-pool system, this heuristic control assigns the same priority orderings as the RPT control when M is set to equal the threshold level L_1. It is reasonable to set the value of M as in Definition 2 (and then potentially adjust its value to achieve better performance). Note that when $J \geq 3$, the priority ordering of the heuristic control and the RPT control do not in general coincide.

Figure 7 compares the RPT controls, the heuristic controls, and the QIR controls in a three-pool system. Figure 7(a) has $T(1,2) \leq T(2,3)$ so pool 2 is reduced; Figure 7(b) has $T(1,2) > T(2,3)$ so no pool is reduced. To generate the figures, we first vary the value of c between 0 and ∞ . Then, for each given c, we solve for the optimal thresholds and generate the plots for the RPT control. We vary M from 0 to ∞ to generate plots for the heuristic control. For

Figure 7 Simulated Comparison Between RPT, Heuristic, and QIR Controls in a Three-Pool System



(a) $\vec{\mu} = (3, 6, 15)$, $\vec{p} = (0.99, 0.8, 0.5)$, $\vec{N} = (25, 25, 25)$, and $\rho = 0.9$ (pool 2 is reduced)

(b) $\vec{\mu} = (3, 6, 15)$, $\vec{p} = (0.99, 0.6, 0.5)$, $\vec{N} = (25, 25, 25)$, and $\rho = 0.9$ (pool 2 is not reduced) the QIR control, we generate (f_1, f_2, f_3) as a three-dimensional grid with even spacing 0.2. More specifically, we choose the ratios from the set $\{(f_1, f_2, f_3) \mid f_1+f_2+f_3=1$ and ; f_j is a multiple of $0.2, j=1,2,3\}$, which results in 21 possible vectors (f_1, f_2, f_3).

We again observe that the RPT control is on the efficient frontier. However, there is a caveat. Recall that Remark 1 notes that any solution to the approximating diffusion control problem generates the entire efficient frontier as c varies from 0 to ∞. However, for λ that is not large enough, there is some additional thought needed when one or more pools are reduced. For example, in Figure 7(a), pool 2 is reduced, and so the RPT control always gives pool 2 the highest priority in routing. Its performance is closest to that of the $p\mu$-rule when its associated threshold is infinite so that the RPT control has static priority ordering $2 \succ 3 \succ 1$. To achieve the same performance as the $p\mu$-rule in a manner that is consistent with the approximating diffusion control problem solution, we "extend" the RPT control using an additional threshold (which we vary from 0 to ∞) that trades off the higher priority pools 2 and 3, keeping pool 1 as the lowest priority pool.[3] This is consistent with the solution to the approximating diffusion control problem because that solution only specifies which pool should remain idle. Finally, we note that this issue does not arise when no pools are reduced, as in Figure 7(b), because then all the thresholds being 0 is consistent with the p-rule, and all the thresholds being ∞ is consistent with the $p\mu$-rule.

The performance of the heuristic control appears to be slightly worse in Figure 7(b) than in Figure 7(a). This is consistent with the observation that the RPT and heuristic controls sometimes give different pools the lowest routing priority in Figure 7(b), but not in Figure 7(a). In particular, the heuristic control either prioritizes the pools as $3 \succ 2 \succ 1$ or $1 \succ 2 \succ 3$. In Figure 7(b), the RPT control sometimes gives pool 2 the lowest priority in routing (as shown in Figure 2(b)); however, in Figure 7(a), pool 2 always has the highest priority in routing, and the lowest priority pool is either 1 or 3 (as shown in Figure 2(a)).

Next, when we compare the QIR and RPT controls, the QIR controls that are closest to the efficient frontier are those having idleness allocations (f_1, f_2, f_3) that are similar to the RPT control. For Figure 7(a), $f_2=0$ matches the idleness allocation of the RPT control. This is because the RPT control always gives pool 2 the highest priority, and so there are rarely idle servers from pool 2. For Figure 7(b), the RPT idleness allocations are as shown in Figure 4(b) in Remark 4. It is interesting to see that the QIR controls with matching idleness allocations sometimes achieve performance very near or on the efficient frontier. This is not surprising around both endpoints of the efficient frontier, because the RPT, QIR, and heuristic controls mimic the p-rule or $p\mu$-rule. However, in general, we cannot expect the RPT and QIR controls to have the same performance, because their resulting diffusion approximations are different. We leave the question of when and how close the performance of the RPT and QIR controls can be for future research.

Finally, we hypothesize that the threshold structure of the RPT control is good, even when the service times are not exponential. First, the identification of the set of pools that should have highest priority in routing (the set $\mathcal{J} - \mathcal{K}^\star$) depends only on the mean service times and resolution probabilities, and not on variance. Hence, it is reasonable to expect that this set of pools is robust to changes in the distributional assumptions on interarrival and service times (even though the approximating DCP relies on the assumption of exponential service times). Second, Figure 8 shows that the RPT control continues to be on the efficient frontier when service times follow a lognormal (not exponential) distribution. Note that the statistical study of a call center in Brown et al. (2005) showed that service times followed a lognormal distribution. We continued to assume Poisson arrivals, which is a simplification of the finding in Brown et al. (2005) that the aforementioned call center had an arrival process that was consistent with a time-varying Poisson process. This can be reasonable in practice, because one way to handle control problems when arrivals are time varying is to split the time horizon into fixed size intervals and assume a constant arrival rate over each interval.

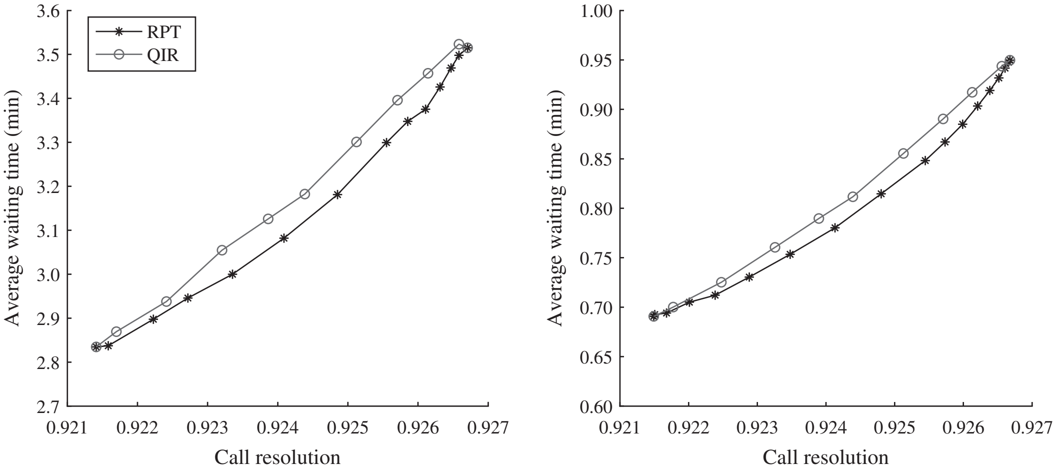
In Figure 8, (a) and (b), we investigate lognormal distributed service times. We simulate two different systems with different coefficients of variation. In Figure 8(a) we assume the service times of pool 1 agents follow the lognormal distribution $F_1 \sim \text{LogNomal}(-2, 1.5^2)$, where if X_1 has distribution F_1, then $\log X_1$ has a normal distribution with mean -2 and variance 1.5^2. The service times of pool 2 agents follow the lognormal distribution $\text{LogNomal}(-3, 1.5^2)$. They both have the same coefficient of variation, which is $\sqrt{\exp(1.5^2)-1}=2.91$. The mean service rates of pool 1 and pool 2 are 2.4 and 6.52, respectively. We vary the threshold from 0 to ∞, which corresponds to varying c from 0 to ∞ to draw the points corresponding to the RPT controls. We vary the f_1 in the QIR control from 0 to 1 to draw the points corresponding to the QIR controls. In Figure 8(b) we let the service times of pool 1 agents follow $\text{LogNormal}(-1, 0.5^2)$, and the service times of pool 2 agents follow $\text{LogNormal}(-2, 0.5^2)$. The mean service rates are still 2.4 and 6.52, respectively, whereas the

[3] Similarly, the performance of the RPT control is closest to that of the p-rule, when its associated threshold is 0, so that the RPT control has static priority ordering $2 \succ 1 \succ 3$. However, pools 1 and 2 are "close enough" that there is no obvious performance difference between the p-rule and RPT control.

Figure 8 Simulated Comparison Between RPT and QIR in a Two-Pool System with Lognormal Service Times

(a) Lognormal service times with CV = 2.91 (b) Lognormal service times with CV = 0.533

$\vec{p} = (0.99, 0.90)$, $\vec{N} = (25, 25)$, $\hat{\rho} = 0.9$



coefficients of variation of the two distributions both change to $\sqrt{\exp(0.5^2)-1} = 0.533$. We see that the RPT controls are still on the efficient frontier. Furthermore, when the coefficient of variation is larger, there is more variation in the average waiting time, but the variation in call resolution is not much affected. (These insights also hold when service times follow a Gamma distribution; see the additional simulation results in Figures EC-1 and EC-2 in the EC.)

6. Conclusions and Future Research

We have proposed a RPT routing control for call centers with homogeneous customers and heterogeneous agents that trades off the dual performance objectives of minimizing the average waiting time and minimizing the callback rate (equivalently, maximizing the call resolution, the overall percentage of arriving calls that are successfully served and doesn't result in callbacks). The pools are reduced because, depending on the system parameters, there may be some pools that always have priority when routing, and so almost never have idle agents. The remaining pools have routing priorities that are dynamically determined using thresholds on the number of idle agents. Then, those pools sometimes have idle agents, and sometimes not, depending on the system state. We have shown that our proposed control is on the efficient frontier with respect to average waiting time and call resolution. We have done this analytically, by solving the relevant approximating diffusion control problem, and through simulation.

The approximating diffusion control problem is analytically tractable because it is one-dimensional. The first important insight coming from the DCP solution is understanding how to reduce the pools, that is, identifying which pools should never have idle agents. We have provided a simple set of parameter conditions (see (21)) to separate out the pools that should never have idle agents. Then, depending on the importance of the customer waiting in relation to call resolution, there may be additional pools that should be reduced. The second important insight coming from the DCP solution is understanding how to dynamically determine the routing priorities for the remaining pools (that have not been reduced). In particular, this determination is made using a control that has a simple threshold structure.

The main practical implication of our work is that focusing on minimizing average waiting time as the sole performance objective may not deliver the best customer experience. In particular, when the performance objective is to minimize the average waiting time (corresponding to c being very large), almost all of the idle agents will be the slowest agents. However, it may be the case that some slow agents have high service quality, because they have high resolution probability. Then, the customer experience suffers, because there is a higher probability that a customer will not have his inquiry properly resolved, and so will call back. Hence, it is important to consider objective functions that take into account call resolution.

Another practical implication of our work concerns staffing. In particular, the RPT control identifies those agent pools that should never be idled. Then, if a call center manager must make a decision regarding which agents to keep and which agents to let go, the agents to keep are those that should never be idled. This leads naturally to a problem formulation that includes the staffing decisions of the system manager, for example, minimizing staffing costs subject to quality of service constraints on the average waiting time and the call resolution.

One interesting direction for future methodological research concerns how to generalize the model to include heterogeneous customers. The identification of the RPT control as the solution to the approximating diffusion control problem for the inverted-V model can be a first step. This is because there are cases in the literature (as mentioned in the introduction) when the approximating diffusion control problem separates into one for a V-model and one for an inverted-V model. There is an additional complication in that the rate at which calls of different types are directed to the different agent pools must first be determined before the approximating diffusion control problem can be formulated, and those rates will affect the system performance.

Our model assumes that service speeds and resolution probabilities are fixed. However, in reality, agents learn (and so, over time, may increase their service speeds and/or resolution probabilities). It is of interest to develop models that incorporate agent learning when agents are heterogeneous. The paper by Arlotto et al. (2014) is one recent work in this direction; however, their focus is on hiring and retention rather than routing.

It is also true that agents make their own individual trade-off decisions between how long to spend with a customer and what service quality to give that customer. This presents the challenge of providing an analytic model for how agents make such decisions that is relevant to the call center environment. Such a model could use routing to incentivize agent behavior, or it could use agent compensation. However, this requires care, because it is easy for agent incentive schemes to backfire. For example, Figure 19 in Gans et al. (2003) documents agents hanging up on customers.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/msom.2013.0463.

Acknowledgments

The authors thank the associate editor and the anonymous reviewers for their constructive suggestions. The authors also thank Li Chen from Duke University for suggesting the heuristic control in Figure 7, and Avi Mandelbaum for helpful discussions related to this paper.

References

Aksin Z, Armony M, Mehrotra V (2007) The modern call-center: A multi-disciplinary perspective on operations management research. *Production Oper. Management* 16(6):655–688.

Alizamir S, de Vericourt F, Sun P (2013) Diagnostic accuracy under congestion. *Management Sci.* 59(1):157–171.

Anand KS, Paç MF, Veeraraghavan S (2011) Quality-speed conundrum: Trade-offs in customer-intensive services. *Management Sci.* 57(1):40–56.

Arlotto A, Chick SE, Gans N (2014) Optimal hiring and retention policies for heterogeneous workers who learn. *Management Sci.* 60(1):110–129.

Armony M (2005) Dynamic routing in large-scale service systems with heterogeneous servers. *Queueing Systems* 51(3–4):287–329.

Armony M, Ward A (2010) Fair dynamic routing in large-scale heterogeneous-server systems. *Oper. Res.* 58(3):624–637.

Atar R (2005) Scheduling control for queueing systems with many servers. *Ann. Appl. Probab.* 15(4):2606–2650.

Atar R (2008) Central limit theorem for a many-server queue with random service rates. *Ann. Appl. Probab.* 18(4):1548–1568.

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

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

Brown L, Gans N, Mandelbaum A, Sakov A, Shen H, Zeltyn S, Zhao L (2005) Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Statist. Assoc.* 100(469):36–50.

Chan CW, Yom-Tov G, Escobar G (2012) When to use speedup? An examination of service systems with returns. Working paper, Columbia University, New York.

de Vericourt F, Zhou Y (2005) Managing response time in a call-routing problem with service failure. *Oper. Res.* 53(6):968–981.

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

Gurvich I, Whitt W (2009a) Queue-and-idleness-ratio controls in many-server service systems. *Math. Oper. Res.* 34(2):363–396.

Gurvich I, Whitt W (2009b) Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing Service Oper. Management* 11(2):237–253.

Halfin S, Whitt W (1981) Heavy-traffic limits for queues with many exponential servers. *Oper. Res.* 29(3):567–588.

Harrison JM, Zeevi A (2004) Dynamic scheduling of a multiclass queue in the Halfin and Whitt heavy traffic regime. *Oper. Res.* 52(2):243–257.

Hart M, Fichtner B, Fjalestad E, Langley S (2006) Contact centre performance: In pursuit of first call resolution. *Management Dynam.* 15(4):17–28.

Kostami V, Rajagopalan S (2014) Speed–quality trade-offs in a dynamic model. *Manufacturing Service Oper. Management* 16(1): 104–118.

Lovejoy WS, Sethuraman K (2000) Congestion and complexity costs in a plant with fixed resources that strives to make schedule. *Manufacturing Service Oper. Management* 2(3):221–239.

Lu L, Van Mieghem J, Savaskan C (2009) Incentives for quality through endogenous routing. *Manufacturing Service Oper. Management* 11(2):254–273.

Mandelbaum A, Momcilovic P, Tseytlin Y (2012) On fair routing from emergency departments to hospital wards: QED queues with heterogeneous servers. *Management Sci.* 58(7):1273–1291.

Mehrotra V, Ross K, Ryder G, Zhou Y (2012) Routing to manage resolution and waiting time in call centers with heterogeneous servers. *Manufacturing Service Oper. Management* 14(1):66–81.

Perry O, Whitt W (2009) Responding to unexpected overloads in large-scale service systems. *Management Sci.* 55(8):1353–1367.

Perry O, Whitt W (2011) A fluid approximation for service systems responding to unexpected overloads. *Oper. Res.* 59(5): 1159–1170.

Tezcan T, Dai J (2010) Dynamic control of N-systems with many servers: Asymptotic optimality of a static priority policy in heavy traffic. *Oper. Res.* 58(1):94–110.

Ward A, Armony M (2013) Blind fair routing in large-scale service systems with heterogeneous customers and servers. *Oper. Res.* 61(1):228–243.

Weerasinghe A, Mandelbaum A (2013) Abandonment versus blocking in many-server queues: Asymptotic optimality in the QED regime. *Queueing Systems* 75(2–4):279–337.