A CALL-ROUTING PROBLEM WITH SERVICE-LEVEL CONSTRAINTS

NOAH GANS

OPIM Department, The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104-6366, gans@wharton.upenn.edu

YONG-PIN ZHOU

Department of Management Science, School of Business Administration, University of Washington, Seattle, Washington 98195-3200, yongpin@u.washington.edu

We consider a queueing system, commonly found in inbound telephone call centers, that processes two types of work. Type-H jobs arrive at rate λ_H, are processed at rate μ_H, and are served first come, first served within class. A service-level constraint of the form $\mathsf{E}[\text{delay}] \leqslant \alpha$ or $\mathbf{P}\{\text{delay} \leqslant \beta\} \geqslant \alpha$ limits the delay in queue that these jobs may face. An infinite backlog of type-L jobs awaits processing at rate μ_L, and there is no service-level constraint on this type of work. A pool of c identical servers processes all jobs, and a system controller must maximize the rate at which type-L jobs are processed, subject to the service-level constraint placed on the type-H work.

We formulate the problem as a constrained, average-cost Markov decision process and determine the structure of effective routing policies. When the expected service times of the two classes are the same, these policies are globally optimal, and the computation time required to find the optimal policy is about that required to calculate the normalizing constant for a simple $M/M/c$ system. When the expected service times of the two classes differ, the policies are optimal within the class of priority policies, and the determination of optimal policy parameters can be determined through the solution of a linear program with $O(c^3)$ variables and $O(c^2)$ constraints.

Received January 2001; revisions received September 2001, November 2001; accepted May 2002.

Subject classifications: Dynamic programming, infinite state Markov: average cost, constrained MDP. Queues, optimization: control of multiclass, Markovian systems.

Area of review: Manufacturing, Service, and Supply-Chain Operations.

1. INTRODUCTION

Consider the following Markovian queueing system that processes two types of work. Type-H work arrives according to a Poisson process of intensity λ_H and has independent and identically distributed (*i.i.d.*), exponentially distributed processing times of mean $1/\mu_H$. Type-H jobs are served first come, first served (FCFS) within class, and they queue for service. A service-level constraint of the form $\mathsf{E}[\text{delay}] \leqslant \alpha$ or $\mathbf{P}\{\text{delay} < \beta\} \geqslant \alpha$ limits the delay in queue that type-H jobs may face. Type-L jobs have *i.i.d.*, exponentially distributed processing times of mean $1/\mu_L$, and there exists an infinite backlog of type-L jobs, so that there is always type-L work to be processed. There is no service-level constraint on type-L work. A system controller routes type-H and type-L jobs to one of c identical servers. The controller's objective is to maximize the rate at which type-L jobs are processed, subject to the service-level constraint placed on type-H jobs.

The primary motivation for analyzing this problem comes from a situation commonly found in inbound telephone call centers. (Inbound calls originate from outside clients calling "in" to a center, while outbound calls originate from employees in a center calling "out.") The problem is one of how best to use pockets of excess capacity, and it originates with a two-step staffing procedure that is embedded in the software packages used to manage many of these centers.

In the first stage of the staffing procedure, the Erlang-C formula (or a variant) is used to determine the minimum numbers of customer service representatives (CSRs) required to be on hand at different times of the day. More specifically, the arrival process is traditionally modeled as a Poisson process in which the rate changes every 30 minutes; within each of these increments the process is considered to be stationary. Service times are assumed to be exponentially distributed. The call center sets a service-level standard for the delay distribution of incoming calls, and the Erlang-C formula is used to find the minimum number of CSRs required to meet the service-level standard in steady state. (For example, see Fleischer 2000, item 29.)

In the second stage, the minimum staffing requirements for these 30-minute increments become the right-hand side of a set-covering integer program (IP) that assigns CSRs to work schedules. Each feasible schedule defines which half-hour increments during the week a CSR will work and which s/he will not. The IP's decision variables are the numbers of CSRs assigned to each of the feasible schedules, and the constraints ensure that the sum of the CSRs working in every half-hour are sufficient to meet the staffing requirements. For example formulations of this mathematical program, see Pinedo et al. (2000) or Gans and Zhou (2002).

Because of limits on the schedules used in the procedure, the IP's solution typically includes half-hour intervals that are overstaffed. That is, the number of CSRs that are available to work exceeds the number required to meet the service-level constraint. During these times, call-center managers often assign lower-priority, postponable work to CSRs. In some organizations, the postponable work may be "outbound" calls, either for sales or to call back customers who have left a message. In other organizations, this work may be responding to emails, processing insurance claims, or performing other clerical work. In our model, inbound

0030-364X/03/5102-0255 $05.00
1526-5463 electronic ISSN

Operations Research © 2003 INFORMS
Vol. 51, No. 2, March–April 2003, pp. 255–271

calls are of type H, postponable work is of type L, and CSRs are servers.

Suppose that during some interval, only $i < c$ CSRs are required to meet the service-level constraint imposed on type-H work. A feasible scheme would be to assign i CSRs to take incoming calls and $(c - i)$ CSRs to handle type-L work. Given the objective of maximizing the throughput of type-L work, however, this scheme is suboptimal; there are times during which all i CSRs dedicated to type-H work may be idle, and at these times some of the i may be used to process type-L work without violating the type-H service-level constraint. Even when $i = c$, additional productive capacity to process type-L jobs may be found by opportunistically routing type-L work at appropriate moments.

Indeed, the latest generation of call-center switches can be programmed to perform these types of routing functions. They handle multiple types of work—inbound calls, outbound calls, emails, and other computer-mediated tasks—and route work to agents based on preprogrammed scripts. For examples of systems that advertise this capability, see Avaya Communications (2001) and Genesys Telecommunications Laboratories (2001). In this paper, we seek to characterize effective work-routing schemes for managing these systems.

We first analyze systems in which $\mu_H = \mu_L$, and we prove that the following stationary scheme is optimal: (1) type-H jobs are given priority over type-L jobs; and (2) type-L jobs are admitted into service according to a randomized "threshold reservation" policy. In a deterministic version of the policy, type-L work is routed to a CSR only when there are no type-H jobs in queue and the number of busy CSRs falls below some fixed threshold number $i \leqslant c$. In the randomized version of the policy, the threshold level is randomized between two adjacent numbers i and $i + 1 \leqslant c$. In this system, the computation required to find the optimal policy is about the same as that required to compute the normalizing constant of an $M/M/c$ queue. Thus, the computational effort to calculate the optimal policy parameters is modest, and the resulting policies are straightforward to understand and implement.

We then evaluate analogous systems for which $\mu_H \neq \mu_L$. In these systems, the amount of work present in the system depends on the composition of the jobs, rather than simply the total number of jobs. In turn, the policies that admit type-L jobs based on the number of busy servers—which were optimal when $\mu_H = \mu_L$—are not necessarily optimal. Here the problem of finding the desired routing policy can be formulated as a linear program (LP) with $O(c^3)$ variables and $O(c^2)$ constraints. This policy is optimal among all those which give priority to type-H jobs.

In this case, performance should continue to be excellent, although the computational effort required to calculate optimal policy parameters grows substantially. For example, for a pool of 100 CSRs, the LP formulation requires roughly 179,000 variables and 5,250 constraints. While this size of problem is well within the capabilites of current LP solvers and can be automated in a straightforward fashion, additional structural results that reduce the scale of the problem are warranted.

In the discussion at the end of the paper, we briefly discuss a class of threshold-type policies that, if implemented, would further reduce the number of variables in the LP by an order of magnitude, so that it would require $O(c^2)$ decision variables and $O(c^2)$ constraints. We conjecture that this subset of policies is optimal (within the class of priority policies).

The remainder of the paper is organized as follows. In §2 we review related literature, and we discuss the assumption that the underlying stochastic process is stationary and Markovian. Then §3 analyzes the case in which $\mu_H = \mu_L$, and §4 the case in which $\mu_H \neq \mu_L$. In §5 we discuss our results as well as directions for future research.

2. RELATED LITERATURE

In this section, we describe two sets of related literature. First, we discuss research articles whose models are similar to the one analyzed in this paper. Next, we discuss papers that address some of the limits inherent in our model's assumption that the underlying system is stationary and Markovian.

The work most closely related to this is that of Bhulai and Koole (2000), who have independently analyzed the same problem. For the case of $\mu_H = \mu_L$, their results match ours. The paper differs from ours in two broad respects, however. First, it uses value iteration as the basis for analysis, while we use an LP formulation of the underlying MDP. Second, for the case of $\mu_H \neq \mu_L$ it analyzes a heuristic that is based on system occupancy, rather than the optimal policy. In contrast, we develop a procedure for finding the optimal routing policy.

Armony and Maglaris (2003) consider a closely related system in which arriving customers are informed of the expected delay and may elect to balk (exit the system upon arrival) or to be called back by a CSR, rather than waiting in queue to be served. In this setting, type-H customers are willing to wait, type-L customers elect to be called back, balking customers are lost, and $\mu_H = \mu_L$. Furthermore, each arriving customer decides which type s/he becomes, based on personal preferences. The objective is to minimize the expected delay of type-H calls, in equilibrium, subject to a delay constraint on type-L calls. The paper demonstrates the asymptotic optimality of a threshold policy in which type-L calls are given priority if and only if the type-L queue exceeds a certain threshold.

Brandt and Brandt (1999) analyze another closely related system in which $\mu_H = \mu_L$. In this system, arriving type-H calls are live customers and are impatient, while type-L calls are "callbacks"—messages left by customers for the call center to return the call—and are patient. After a random amount of time spent waiting in queue, the impatient, type-H calls may turn into type-L calls or abandon (exit) the system without being served. Given a fixed threshold-reservation policy, the paper develops a system of integral

equations that characterizes the steady-state distribution of the numbers being served and in the type-H queue. It also develops an approximate characterization of the steady-state number in the type-L queue.

These two papers differ from ours in three respects. First, the models developed in Armony and Maglaras (2003) and Brandt and Brandt (1999) seek to characterize systems in which there is not enough capacity and underserved type-H customers do not wait in queue. The model we present, however, aims to characterize a complementary situation in which there is ample capacity for type-H calls and little or no abandonment by type-H callers. Rather, the primary question is how best to use excess capacity without adversely affecting the service level of type-H calls. Second, the dynamics of the systems analyzed in the two papers differ somewhat from those of the system analyzed in our paper. Finally, the two papers do not address cases in which $\mu_H \neq \mu_L$.

In the telecommunication literature, there are also routing control problems similar to the system described in this paper and, again, the models typically assume that $\mu_H = \mu_L$. Blanc et al. (1992) describe a system in which type-L jobs arrive according to a Poisson process and are subject to admission control. Once admitted, type-L jobs queue together with type-H jobs, and the objective is to maximize a discounted total reward associated with the number of jobs accepted. A "threshold" type policy—in which type-L calls will be admitted only when the total calls in the system (type-H plus admitted type-L calls) does not exceed a fixed limit—is proved to be optimal. Guérin (1988) models a cell phone network in which type-L calls are those that originate from within the current cell and type-H calls are those being handed off from another cell. Because type-H calls are already in process, if they are not immediately served upon arrival they are lost, and a limit is imposed upon their loss probability. For this system, the paper analyzes policies for reserving channels that are analogues of the threshold reservation schemes in our paper. Guérin (1988) does not address the optimality of this class of policies, however.

Berman and Larson (2000) consider retail operations in which type-H work takes the form of customers arriving to cash registers, and type-L work awaits in a "back room" in which restocking and other maintenance activities must be performed. In this model, back room work does not consist of a number of separate jobs, but rather a total quantity of work that must be attended to each period, and switching time is incurred each time servers change from customer checkout to back-room tasks. The paper proposes two heuristics for managing servers that are hysteresis versions of threshold reservation policies. The paper does not address the optimality of these policies, however.

Earlier work by Schaack and Larson (1986) uses generating functions to characterize the performance of threshold reservation policies for systems in which $m \geqslant 2$ classes of customers, all with the same, exponential service time distribution. Again, the paper concentrates on performance analysis and does not introduce a notion of optimality as it analyzes policies. In addition, it does not explicitly address service-level constraints, and it does not characterize systems in which service time distributions may vary among customer classes.

Carr and Duenyas (2000) study a job admission and sequencing problem for a single-server queue in the inventory/production setting. Their system has two streams of jobs with different service standards.

Akşin and Harker (2001) consider the impact of the addition of background work to congestion in the information systems that support the agents that handle calls. They model this effect using a processor-sharing loss system.

The standard call-center staffing procedure described in the introduction models the arrival and service process of type-H calls as that of a stationary $M/M/c$ queue. While these assumptions are not wholly correct, we adopt them in the interest of analytical tractability. We address each of them in turn.

The arrival process at many call centers is nonstationary, and this process may be affected by various factors, such as line of business, geography, type of technology in use. Recent empirical work by Brown et al. (2002) characterizes the arrival process at one inbound call center as a nonstationary Poisson process. There is also stream of research that addresses the fact that standard procedures "discretize" nonstationarities in the arrival process, and they attempt to adjust staffing levels to reflect nonstationarities within half-hour increments. For example see Jennings et al. (1996), Green et al. (2000), and the references therein.

Similarly, much call-center software and research has assumed that the service-times of calls are exponentially distributed. In particular, there is an active stream of research, based on earlier work by Halfin and Whitt (1981), that uses asymptotic analysis for systems with many exponential servers. For example, see Borst et al. (2000) and Garnett et al. (2002). Recent empirical evidence, however, suggests that service times may not be exponentially distributed. In particular, Bolotin (1994) and Mandelbaum et al. (2000) have both found the duration of talk times to be lognormally distributed. To capture effects such as these, Puhalskii and Reiman (2000) have recently generalized the original, asymptotic analysis of Halfin and Whitt to cover the class of GI/PH/c queues.

3. IDENTICAL SERVICE-TIME DISTRIBUTIONS: $\mu_H = \mu_L$

In this section we analyze systems in which the service-time distributions of the two types of calls are identical: $\mu_H = \mu_L \equiv \mu$.

3.1. Model

Again, we assume that type-H jobs arrive according to a Poisson process of intensity λ_H, that there exists an infinite backlog of type-L jobs and that service times are exponen-

tially distributed with rate $\mu_H = \mu_L \equiv \mu$. Let $\rho = \lambda_H / c\mu_H$ be the offered load of type-H jobs. Clearly, whenever $\rho < 1$ there exist stable policies.

The fact that interarrival times and service times are exponentially distributed implies that, rather than analyzing the continuous time Markov chains (CTMCs) induced by control policies, we can uniformize the event rate of the CTMC to analyze an equivalent discrete time Markov chain (DTMC) that is embedded at a uniformized set of event epochs (see Puterman 1994). In particular, uniformization ensures that the transition rate out of any state is always the same, so that the fraction of time the CTMC spends in a given state corresponds exactly to the fraction of transitions that the embedded (after action) DTMC spends in that state.

We let the uniformization rate equal $\lambda_H + c\mu$. Furthermore, without loss of generality, we define the time scale so that $\lambda_H + c\mu = 1$. Therefore, we may view transition rates as probabilities. For example, $\lambda_H = \lambda_H/(\lambda_H + c\mu)$ equals the expected number of type-H arrivals per period, as well as the probability that the next event is a type-H arrival.

Because $\mu_H = \mu_L \equiv \mu$ the state of the system at discrete event epoch t can be described using two dimensions: the number of busy servers i, and the number of type-H calls in queue q. We define the state space to be $S = \{(i, q)\colon 0 \leqslant i \leqslant c,\ \ q \geqslant 0\}$ and $s_t \in S$ to be the state of the system at event epoch t, *before* any action is taken.

In any state, a system controller may put one or more calls into service, or it may do nothing. Accordingly, let j and k be the numbers of type-H and type-L calls put into service at an arbitrary event epoch. We define the set of feasible actions in state $s \in S$ to be $A_s = \{(j, k)\colon 0 \leqslant j \leqslant \min\{c - i, q\},\ 0 \leqslant k \leqslant c - i - j\}$, as well as the action taken at time t to be $a_t \in A_{s_t}$. We denote the superset of all feasible actions as $A = \{(j, k)\colon 0 \leqslant j \leqslant c,\ 0 \leqslant k \leqslant c,\ j + k \leqslant c\} \supseteq A_s$ for all $s \in S$. Observe that A is finite.

Because of the correspondence between the CTMC and the after-action DTMC, at times it will be more convenient to consider the state of the system *after* an action is taken, rather than before. Accordingly, we define the after-action state space, $\bar{s}_t \in \overline{S} = \{(\bar{\imath}, \bar{q})\colon 0 \leqslant \bar{\imath} \leqslant c,\ \bar{q} \geqslant 0\}$, to be the system state after an action is taken at event epoch t.

The relationship between $\bar{s}_t$ and s_{t+1} is as follows:

$$(i_{t+1}, q_{t+1}) = \begin{cases} (\bar{\imath}_t, \bar{q}_t + 1), & \text{w. p. } \lambda_H; \\ (\bar{\imath}_t - 1, \bar{q}_t), & \text{w. p. } \bar{\imath}_t \mu; \text{ and} \\ (\bar{\imath}_t, \bar{q}_t), & \text{w. p. } (c - \bar{\imath}_t)\mu. \end{cases} \tag{3.1}$$

Two points are worth noting. First, when $\bar{\imath}_t = 0$, the second transition, to $(\bar{\imath}_t - 1, \bar{q}_t)$, occurs with probability zero. Second, the transition in which $(i_{t+1}, q_{t+1}) = (\bar{\imath}_t, \bar{q}_t)$ is the result of uniformization at rate $\lambda_H + c\mu$.

A *policy* is a set of decision rules that the system controller uses when choosing an action to take at each event epoch t. Define the *history* of the system up to event epoch t to be $\mathcal{H}_t = \{(s_0, a_0), \ldots, (s_{t-1}, a_{t-1}) \cup s_t\}$, the record of all states entered and actions taken up through event epoch t. A *nonanticipating* policy is a rule that, given $\mathcal{H}_t$, chooses an action a_t, possibly at random, among the actions of A_{s_t}. We consider only such nonanticipating policies.

Furthermore, we limit ourselves to the class of policies that stabilizes the expected number of type-H calls in the backlog. This eliminates from consideration policies that on certain sample paths may let the expected backlog grow without bound.

More formally, we define $\Lambda_H(n)$ to be the number of type-H arrivals over the first n transitions. Similarly, we let $\overline{H}_\pi(n)$, $\overline{L}_\pi(n)$, and $\bar{n}_\pi(n) = \overline{H}_\pi(n) + \overline{L}_\pi(n)$ be the (after action) numbers of type-H, type-L, and all jobs put into service by policy π through event epoch n. In turn, we let $\bar{q}_\pi(n) = \Lambda_H(n) - \overline{H}_\pi(n)$ be the number of type-H jobs in the queue after the first n transitions. Because it is bounded by $\Lambda_H(n)$, the expected queue length must grow at most linearly under any policy: $\limsup_{n\to\infty} \mathsf{E}[\bar{q}_\pi(n)]/n \leqslant \lambda_H$. A stabilizing policy further restricts this limit to be zero.

DEFINITION 1. Let Π denote the class of *admissible* policies. These are nonanticipating policies for which the limit $\lim_{n\to\infty} \mathsf{E}[\bar{q}_\pi(n)]/n$ exists and equals zero.

Note that $\overline{H}_\pi(n) = \Lambda_H(n) - \bar{q}_\pi(n)$ and that

$$\lim_{n\to\infty} \mathsf{E}[\Lambda_H(n)]/n = \lambda_H.$$

Therefore, we must have

$$\lim_{n\to\infty} \mathsf{E}\left[\overline{H}_\pi(n)\right]/n = \lambda_H, \quad \forall\ \pi \in \Pi, \tag{3.2}$$

as well.

Thus, we seek an optimal policy π among the class of admissible policies, Π. A *stationary* policy takes the same (possibly randomized) action $a_t \in A_{s_t}$, based only on s_t and not the previous history through $(t-1)$. A stationary policy induces a Markov chain on S. We will show that there exists a stationary policy $\pi \in \Pi$ that is optimal.

The objective is to maximize the rate at which type-L calls are served. For a given state s define the *reward* associated with action a to be $R(s, a)$. We let $R(s, a) = k$, the number of type-L jobs put into service. In turn, we define

$$\overline{R}_\pi(s) \stackrel{\text{def}}{=} \liminf_{n\to\infty} \frac{1}{n} E_\pi \left[\sum_{t=0}^{n-1} R(s_t, a_t) \,\middle|\, s_0 = s \right] \tag{3.3}$$

to be the long-run average rate at which a policy $\pi \in \Pi$ serves type-L calls.

To account for the service-level constraint, we denote by $D(s, a)$ the "delay cost" associated with state-action combination (s, a), and we let $D(s_t, a_t) = d(\bar{q}_t)$ for some nonnegative function of the after-action state of the type-H queue, $\bar{q}_t$. In turn, we define

$$\overline{D}_\pi(s) \stackrel{\text{def}}{=} \limsup_{n\to\infty} \frac{1}{n} E_\pi \left[\sum_{t=0}^{n-1} D(s_t, a_t) \,\middle|\, s_0 = s \right], \tag{3.4}$$

and we require that $\overline{D}_\pi(s) \leqslant D^*$, where D^* is an exogenously defined upper bound on the average backlog cost. This is the service-level constraint.

Two facts concerning $D(s, a)$ are important to note. First, $D(s, a)$ and $\overline{D}_\pi(s)$ are defined as functions of queue occupancy, rather than delay in queue. Given the form of the class of optimal policies, however, we will be able to show that common versions of occupancy and delay constraints are equivalent. We do this in §3.5. Second, the form of $d(\cdot)$ may vary with the particular type of service-level constraint desired. For example, for a constraint on the average number of type-H calls in queue we let $d(\bar{q}_t) = \bar{q}_t$, and for a limit on the probability of the type-H queue exceeding length q^*, we let $d(\bar{q}_t) = 1$ if $\bar{q}_t > q^*$ and zero otherwise.

To maintain the analytical tractability of the problem, we impose the following mild set of restrictions on the form of $d(\cdot)$.

ASSUMPTION 1. (i) $d(0) = 0$ *and* $d(\bar{q})$ *is nondecreasing in* $\bar{q}$;

(ii) $\sup_{\bar{q}} d(\bar{q}) > D^*$; *and*

(iii) $\tilde{d}(\alpha) \stackrel{\text{def}}{=} \sum_{\bar{q}=0}^{\infty} \alpha^{\bar{q}}\, d(\bar{q}) < \infty$ *for all* $\alpha \in (0, 1)$.

Item (i) implies that the cost never decreases as the backlog grows. Together items (i) and (ii) are sufficient to ensure that any sample path for which $\lim_{n\to\infty} \overline{H}(n)/n < \lambda_H$ is also one for which $\lim_{n\to\infty} d(\bar{q})/n > D^*$. Finally, item (iii) defines the generating function $\tilde{d}$ and implies that the service-level cost of occupancy grows subexponentially. All these restrictions are satisfied by formulations of standard service-level constraints, such as bounds on expected occupancy and on the probability of occupancy exceeding a prespecified limit.

Given these definitions of policy, reward, and cost, we can state the system controller's problem as

$$\sup_{\pi\in\Pi} \quad \overline{R}_\pi(s) \quad \text{s.t.} \quad \overline{D}_\pi(s) \leqslant D^*. \tag{3.5}$$

Problem (3.5) is called a *constrained optimization problem* (COP). Any policy π that satisfies the constraint in Problem (3.5) is called *feasible*. If it also achieves the supremum in Problem (3.5), then it is *constrained optimal* (or, it solves the COP). To characterize the form of an optimal class of policies, we follow Sennott (2001), who uses Lagrangian relaxations of the COP to construct a stationary, randomized policy that is optimal.

For finite $|S|$ and $|A|$ the general idea works as follows. First, it is well known that stationary policies for MDPs may be computed using LPs in which the decision variables are the relative frequencies associated with the state-action pairs $\{(s, a) \mid s \in S,\ a \in A_s\}$. Furthermore, each deterministic policy corresponds to a *basic* solution to the corresponding LP. Because exactly one action is taken in each state, the rank of the basis is $|S|$. Second, consider the LP associated with a stationary, deterministic policy that is optimal. By introducing a single service-level constraint, the COP forces the basis of the LP to grow to rank $|S| + 1$, and there must be one state in which an optimal policy randomizes its action. Equivalently, one may think of this optimal policy as a randomization between two stationary, deterministic policies whose actions are the same in every state except one. While in our case $|S|$ is infinite, the same intuition holds.

3.2. Optimality of Type-H Priority Policies

As a first step in the analysis, we use a coupling argument to prove the optimality of priority policies. This optimality provides some insight into what makes a routing policy effective: priority systems route the "important" jobs first, and the proof identifies that the important jobs in our system are the ones for which there exists a service-level constraint. Furthermore, the system behavior that results from priority policies will allows us to apply the LP-based intuition described in §3.1 in a straightforward way to derive additional, important structural results.

More formally, we define priority and work-conserving policies as follows.

DEFINITION 2. A type-H *priority* policy never puts type-L jobs into service when there is a type-H job in queue.

DEFINITION 3. A type-H *work-conserving* policy always serves waiting type-H calls whenever there are idle CSRs.

Given these definitions we have the following lemma.

LEMMA 1. *If there exists a feasible policy,* $\pi \in \Pi$, *then there exists a type-H priority, type-H work-conserving policy that is optimal.*

PROOF. Please see the appendix. □

Two assumptions are central to the lemma's proof. First, the fact that $\mu_H = \mu_L$ allows us to couple the service time of a type-L job put into service in one (nonpriority, nonwork-conserving) system with that of a type-H job put into service in an alternative system that satisfies Definition 2. This, in turn, implies that the alternative policy will be feasible. Second, the fact that there is always a type-L job waiting to be served implies that, by routing type-H jobs ahead of type-L jobs, no type-L jobs will be lost. This assures that the alternative policy will be optimal.

Thus, the lemma shows that it should be possible to give priority to type-H calls without hurting the throughput rate of type-L work. This implies that the search for optimal policies—which opportunistically put type-L work into service—may be restricted to policies that put type-H calls into service whenever a CSR becomes available.

The lemma has important computational implications. If a policy gives priority to and is work conserving with respect to type-H calls, then it must be the case that $\bar{\imath} < c \Rightarrow \bar{q} = 0$ and $\bar{q} > 0 \Rightarrow \bar{\imath} = c$. This implies that we may model the state-space as one dimensional.

To do so, we let $S = \{0, 1, \dots\}$. In $s \in \{0, 1, \dots, c\}$, $s = i$ CSRs are busy and no type-H jobs are in queue, and in $s > c$, all c servers are busy and $q > 0$ type-H jobs are in queue. Note that at event epoch 0 the system may have $i < c$ servers busy and $q > 0$ type-H jobs in queue. Because every type-H priority policy requires that $\min\{q, c - i\}$ type-H calls immediately be put into service, however, without loss of generality we may also assume that at epoch 0 the system's before action state falls within the reduced form of S.

We then simplify the state transition equations (3.1) to reflect the priority of type-H jobs:

$$s_{t+1} = \begin{cases} \bar{s}_t + 1, & \text{w. p. } \lambda_H; \\ \bar{s}_t - 1, & \text{w. p. } \bar{\imath}_t \mu; \text{ and} \\ \bar{s}_t, & \text{w. p. } (c - \bar{\imath}_t)\mu. \end{cases} \tag{3.6}$$

Note that, while the notation has changed, only the first transition equation's dynamics change from Equation (3.1) to Equation (3.6). The new equation reflects the fact that, when a type-H job arrives, if a server is available, it is immediately put into service.

Given a type-H priority, work-conserving scheme, the only before-action states in which the optimal action is not well defined are those for which $i < c$. In these c states one must decide whether to put one or more type-L jobs into service or to do nothing, and the resulting decision problem is far simpler that the general problem of finding an optimal policy. Over the c states, there are $(c+1)(c+2)/2 - 1$ possible actions that the system may take: routing c or fewer type-L jobs when the system is empty, routing $c-1$ or fewer type-L jobs when one server is busy, and so forth.

We modify the action space accordingly. For before-action state s, we let $A_s = \{0, \ldots, (c-s)^+\}$, where $a \in A_s$ represents the number of type-L jobs put into service and $(\cdot)^+ = \max\{\cdot, 0\}$. When $s < c$, as many type-L jobs may be routed as there are idle CSRs. When $s \geqslant c$ and all c CSRs are busy, however, $A_s = \{0\}$, and no type-L jobs can be routed.

Finally, we note that because no action may be taken in states $s \geqslant c$, the portion of the DTMC governing system evolution in these states is fixed. Furthermore, the evolution of the system in these states is that of a birth and death process with birth rate λ_H and death rate $c\mu$. While the set of these states is infinite, we can use simple algebraic substitution to develop closed-form expressions for essential quantities related to them. This allows us to formulate the problem of finding an optimal policy as the solution of a finite-dimensional LP with $O(c^2)$ variables and $O(c)$ constraints.

3.3. Analysis of δ-Optimal Policies

Sennott (2001) shows that in certain cases one may use Lagrangian relaxations of the COP to identify two stationary, deterministic policies that should be randomized, and she provides a method of implementing the scheme using value iteration. In this section we study the Lagrangian relaxation and develop essential structural properties for the relaxed version of our specific COP. These properties provide further insight into the nature of the optimal control, and they provide the basis for a computationally efficient policy-iteration method for solving the COP.

We define the Lagrangian relaxation as follows. Given a fixed Lagrange multiplier $\delta \in [0, \infty)$ let

$$F_\delta(s,a) \stackrel{\text{def}}{=} R(s,a) - \delta D(s,a), \qquad \forall\ s \in S, a \in A_s, \tag{3.7}$$

be a new one-period reward function with associated long-run average:

$$J_{\pi,\delta}(s) = \liminf_{n\to\infty} \frac{1}{n} E_\pi \left[\sum_{t=0}^{n-1} F_\delta(s_t, a_t) \middle| s_0 = s \right] \geqslant \overline{R}_\pi(s) - \delta \overline{D}_\pi(s), \quad \forall\ s \in S. \tag{3.8}$$

In turn, we define

$$J_\delta(s) \stackrel{\text{def}}{=} \sup_\pi J_{\pi,\delta}(s), \quad \forall\ s \in S. \tag{3.9}$$

DEFINITION 4. A policy $\pi \in \Pi$ is *δ-optimal* if $J_\delta(s) = J_{\pi,\delta}(s) = \overline{R}_\pi(s) - \delta \overline{D}_\pi(s),\ \forall s \in S$.

By definition, a δ-optimal policy will have to achieve the supremum in Equation (3.9) as well as the equality in Equation (3.8).

Note that in the Lagrangian relaxation, the objective function is modified and the service-level constraint is eliminated. The system dynamics under any policy remain the same, however. For this relaxed problem, we demonstrate that there exists a member of the following class of (stationary) policies that is optimal.

DEFINITION 5. A *generalized threshold reservation* policy with parameters $i^* \in \{0, \ldots, c-1\}$ and $a^* \in \{0, \ldots, c - i^*\}$ puts a^* type-L jobs into service when and only when the system enters state i^*.

Given any stationary policy $\pi \in \Pi$, let $\xi_\pi(s,a)$ be the stationary probability of being in state s and taking action a. That is, fix a stationary policy π, so that in every state s a (possibly randomized) action is fixed. Let M_π be the DTMC induced by π, so the vector ξ_π is the solution to $\xi = \xi M_\pi$, $\xi \mathbf{1} = 1$. Then we have Lemma 2.

LEMMA 2. *Suppose $\rho < 1$. Then there exists a stationary, deterministic policy that is δ-optimal. Furthermore, each stationary deterministic policy π:*

(i) *induces a single, positive recurrent class of states, and the expected absorption time into that class is finite*;

(ii) *has limiting state-action frequencies which correspond to the stationary distribution of the induced Markov chain*: $\lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} \mathbf{1}\{(s_t, a_t) = (s,a)\} = \xi_\pi(s,a)$ *w.p.* 1 $\forall\ s_0$;

(iii) *has uniformly integrable one-period revenues*:

$$\sum_{s\in S, a\in A_s} \xi_\pi(s,a) |R(s,a) - \delta D(s,a)| < \infty; \text{ and}$$

(iv) *corresponds to a generalized threshold reservation policy.*

PROOF. Please see the appendix. □

Parts (i)–(iii) of the lemma allow us to apply Altman and Schwartz's (1991) Theorem 7.1 to demonstrate that there is a one-to-one correspondence between stationary policies and feasible solutions to an appropriately defined infinite-dimensional LP. The lemma's main statement—that there exists a stationary policy that is δ-optimal—ensures that the optimal solution to the LP finds a δ-optimal policy.

REMARK 1. The lemma's proof that there exists a stationary, deterministic policy that is δ-optimal reduces the formulation of the infinite-state-space MDP to that of a finite-state-space semi-Markov decision process (SMDP). In the SMDP, states $c+1, c+2, \ldots$ are eliminated, and the expected length of time and revenue associated with visits to state c are functions of the "busy-period" behavior of the system each time it hits state c. Indeed, for systems with $\mu_H = \mu_L$ it is not difficult to explicitly calculate the statistics associated with visits to state c and to directly solve the SMDP. When $\mu_H \neq \mu_L$, however, the calculations become difficult, and an approach that formulates and solves the associated infinite-dimensional LP becomes preferable. This latter approach is the one we take in this section as well.

To formulate the LP we define decision variables $\xi(s,a)$ that correspond to the state-action frequencies associated with a randomized, stationary policy and let $K_i \stackrel{\text{def}}{=} \{(i',a) \mid i' \in S,\ a \in A_{i'},\ i'+a=i\}$ be the set of state-action pairs (i',a) that takes the system from state i' to state i. Then we have

$$\max \quad \sum_{s=0}^{c-1}\sum_{a=0}^{c-s} a\xi(s,a) - \delta \sum_{s=c+1}^{\infty} d(s-c)\xi(s,0), \tag{3.10}$$

$$\text{s.t.} \quad \sum_{a=0}^{(c-s)^+} \xi(s,a) = \lambda_H \sum_{(i',a)\in K_{s-1}} \xi(i',a) + (c-s)^+\mu \sum_{(i',a)\in K_s} \xi(i',a) + \min\{(s+1),c\}\mu \sum_{(i',a)\in K_{s+1}} \xi(i',a) \qquad s \in \{0,\ldots,c+1\}, \tag{3.11}$$

$$\xi(s,0) = \lambda_H \xi(s-1,0) + c\mu\xi(s+1,0) \qquad s \in \{c+2, c+3, \ldots\}, \tag{3.12}$$

$$\sum_{s=0}^{\infty}\sum_{a=0}^{(c-s)^+} \xi(s,a) = 1, \tag{3.13}$$

$$\xi(s,a) \geqslant 0 \quad \forall\ s \in S,\ a \in A_s. \tag{3.14}$$

Note that whenever $K_i = \varnothing$ we define the associated summation in Equation (3.11) to equal zero.

The LP can be interpreted as follows. The objective function (3.10) is the average number of type-L jobs put into service per period minus the δ-cost of the average backlog per period. The $c+2$ constraints (3.11) represent balance equations that include states in which one or more type-L jobs may be put into service, and the infinite set of constraints (3.12) represents the balance equations for states in which all c CSRs are busy. The left-hand sides of Equations (3.11) and (3.12) are the flows out of each of the states, and the right-hand sides are the flows in: from one state below (if there is one), from the same state (due to uniformization), and from one state above. Constraint (3.13) ensures that the state-action frequencies sum to one.

Observe that $\sum_{a=0}^{c} \xi(c-a,a)$ equals the steady-state distribution of being in state c after action. Similarly, because only action $a=0$ is available in states $s > c$, $\xi(s,0)$ represents the steady-state distribution on being in state $s > c$ after action (as well as before action).

Therefore, the substitution $\xi(s,0) = \sum_{a=0}^{c} \xi(c-a,a)\rho^{s-c}$ satisfies the balance equations for states $s \geqslant c+1$ in which there is a positive queue length. In particular, for $s = c+1$ we may substitute for Equations (3.11), and for $s \geqslant c+2$ we may substitute for all the balance Equations (3.12). In turn, this implies that we may collect terms to write the second summation in the maximand of the objective function as

$$\begin{aligned}
&\delta \sum_{s=c+1}^{\infty} d(s-c)\,\xi(s,0) \\
&\quad = \delta \sum_{s=c+1}^{\infty} \left[d(s-c) \sum_{a=0}^{c} \xi(c-a,a)\rho^{s-c} \right] \\
&\quad = \delta \tilde{d}(\rho) \sum_{a=0}^{c} \xi(c-a,a).
\end{aligned} \tag{3.15}$$

Here $\tilde{d}(\rho)$ is defined as in Assumption 1. Similarly, we may simplify the left-hand side of constraint (3.13) as follows:

$$\begin{aligned}
&\sum_{s=0}^{\infty}\sum_{a=0}^{(c-s)^+} \xi(s,a) \\
&\quad = \sum_{s=0}^{c}\sum_{a=0}^{c-s} \xi(s,a) + \sum_{s=c+1}^{\infty}\sum_{a=0}^{c} \xi(c-a,a)\rho^{s-c} \\
&\quad = \sum_{s=0}^{c}\sum_{a=0}^{c-s} \xi(s,a) + \sum_{a=0}^{c} \xi(c-a,a)\frac{\rho}{1-\rho}.
\end{aligned} \tag{3.16}$$

Then, having eliminated the $\xi(s,0)$, $s > c$ terms from Equations (3.12) and (3.13), we may also eliminate the balance constraints (3.12).

Thus, rather than solving the original, infinite-dimensional LP, we may solve the following finite LP—with $(c+1)(c+2)/2$ decision variables and $c+1$ constraints—to find a δ-optimal policy

$$\max \quad \sum_{s=0}^{c-1}\sum_{a=0}^{c-s} a\xi(i,a) - \delta\tilde{d}(\rho) \sum_{a=0}^{c} \xi(c-a,a), \tag{3.17}$$

$$\text{s.t.} \quad \sum_{a=0}^{c-s} \xi(s,a) = \lambda_H \sum_{(i',a)\in K_{i-1}} \xi(i',a) + (c-s)\mu \sum_{(i',a)\in K_i} \xi(i',a) + (s+1)\mu \sum_{(i',a)\in K_{i+1}} \xi(i',a) \qquad s \in \{0,\ldots,c-1\}, \tag{3.18}$$

$$\sum_{s=0}^{c}\sum_{a=0}^{c-s} \xi(s,a) + \sum_{a=0}^{c} \xi(c-a,a)\frac{\rho}{1-\rho} = 1, \tag{3.19}$$

$$\xi(s,a) \geqslant 0 \quad \forall\ s \in S,\ a \in A_s. \tag{3.20}$$

Note that the formulation does not include a constraint (3.18) in which $\xi(c, 0)$ appears on the left-hand side. This reflects the fact that LP formulations of average cost MDP's are overconstrained, and one balance constraint may be dropped (see §8.8 in Puterman 1994).

In addition, the fact that the problem can be formulated as a finite-state LP allows us to exploit the close relationship between basic solutions of LPs and stationary, deterministic policies for MDPs. By further analyzing the problem in this context, we can demonstrate the δ-optimality of threshold reservation policies that put at most one type-L job into service at a time.

Recall that the class of generalized threshold reservation policies is δ-optimal. We demonstrate that, for any such policy with threshold i^* and routing number $a^* > 1$, there exists an alternative, randomized (nonthreshold) policy with the same performance that routes at most one type-L job at a time:

LEMMA 3. *Given a generalized threshold reservation policy with parameters* (i^*, a^*) *where* $a^* > 1$, *there exists an alternative* (*randomized*) *policy with the same performance that routes at most one job in each of the states* $i^*, i^*+1, \ldots, i^*+a^*-1$.

PROOF. Please see the appendix. □

Therefore, in the search for δ-optimal policies we can restrict our attention to policies that, though they may randomize, route at most one type-L job at a time. In the context of the LP, we may eliminate all decision variables $\xi(s, a)$ for which $a > 1$.

Now consider an optimal, basic feasible solution to a reduced version of the LP (3.17)–(3.20) in which $A_s = \{0, 1\}$ for all $0 \leqslant s \leqslant c-1$. This solution corresponds to a stationary, deterministic policy. Furthermore, given Lemma 3, we know that this policy corresponds to an optimal solution of the original LP (3.17)–(3.20) in which $A_s = \{0, \ldots, c-s\}$ for all $0 \leqslant s \leqslant c-1$. Thus, from part (iv) of Lemma 2 we know that this policy is of the generalized threshold reservation type. Thus, we have proved the following lemma.

LEMMA 4. *If the LP* (3.17)–(3.20) *is feasible, then there exists a generalized threshold reservation policy with threshold* i^* *and routing number* $a^* = 1$ *that is* δ*-optimal.*

We call the threshold reservation policy with $a^* = 1$ "simple" and define these simple policies as follows.

DEFINITION 6. A *simple threshold reservation policy* with threshold $\bar{i}^* = i$ puts one type-L job into service whenever the (before action) system state falls to $i-1$.

Note that in generalized threshold reservation policies i^* is the before action system state, and the after action state equals i^*+a^*. In these simple threshold policies, however, $\bar{i}^*$ is an after-action state—the number in system after the type-L job has been routed. This use of the after-action state to index the policy parameter will make more straightforward the discussion of which $\bar{i}^*$ is optimal.

To determine a δ-optimal policy, we need only evaluate at most $c+1$ of these simple threshold reservation policies—with thresholds from 0 to c. Furthermore, each threshold reservation policy is straightforward to evaluate. Because there is always a type-L job waiting to be served, after-action states for which the number of busy servers fall below $\bar{i}^*$ are transient, and a policy with threshold $\bar{i}^*$ is the analogue of an $M/M/c$ system in which $\bar{i}^*$ servers are always busy. By working from $\bar{i}^* = 0$ to $\bar{i}^* = c$, we may evaluate all $c+1$ policies at once. In the following section, in which we prove that randomized versions of these policies are constrained-optimal, we make the calculations explicit.

3.4. Solution to the COP

We are now ready to complete our characterization of the solution of the COP. As in Sennott (2001), we use the following relationship between δ-optimal policies and the COP to construct an optimal solution to the latter.

LEMMA 5 (BEUTLER AND ROSS 1985). *If for some fixed* $\delta \in [0, \infty)$ *there exists a* δ*-optimal policy* π *such that, for all* $s \in S$, $\overline{R}_\pi(s) < \infty$ *and* $\overline{D}_\pi(s) = D^*$, *then* π *is constrained optimal.*

From §3.3 we know the form of δ-optimal policies. To use Lemma 5, however, we need to calculate the $\overline{R}_\pi(s)$'s and $\overline{D}_\pi(s)$'s associated with each threshold level $\bar{i}^*$. Because we consider only stationary, deterministic policies $\overline{R}_\pi(s) \equiv \overline{R}_\pi$ and $\overline{D}_\pi(s) \equiv \overline{D}_\pi$ for all $s \in S$. For a policy with reservation threshold $\bar{i}^* = i$, we denote $\overline{R}_\pi$ by $\overline{R}_i$ and $\overline{D}_\pi$ by $\overline{D}_i$.

Given $\rho < 1$ and a fixed threshold $\bar{i}^* = i$, we know that states $s \geqslant i$ are positive recurrent and states $s < i$ are transient. We again modify the definition of the state space and transition equations to reflect the simplified system dynamics. For threshold $\bar{i}^* = i$ let $S_i = \{i, i+1, \ldots\}$. Similarly, given a fixed $\bar{i}^* = i$, there are no additional actions to be determined, and without loss of generality we redefine the transition from epoch t to $t+1$ to take place between before-action states:

$$s_{t+1} = \begin{cases} s_t + 1, & \text{w. p. } \lambda_H; \\ \max\{s_t - 1, i\}, & \text{w. p. } \min\{s_t, c\} \cdot \mu; \text{ and} \\ s_t, & \text{w. p. } (c - s_t)^+ \mu. \end{cases} \quad (3.21)$$

Finally, for a fixed simple threshold $\bar{i}^* = i$ we let $\xi_i(s)$ be the stationary probability that the system is in state $s \in S_i$. That is, because the policy is already fixed, the state s refers to that of the Markov chain induced by policy $\bar{i}^* = i$. (Alternatively, we may think of $\xi_i(s)$ as being a slight abuse of notation in which the subscript i simultaneously defines the actions in all states and the state s refers to an after-action state.)

Analysis of the balance equations induced by Equation (3.21) allows us to determine the $\xi_i(s)$s, and in turn

$\overline{R}_i$ and $\overline{D}_i$, in a straightforward fashion. More specifically, for states $i \leqslant s < c$ we have

$$\xi_i(s) = \left(\frac{\mu}{\lambda_H}\right)^{c-s} \left(\frac{c!}{(c-s)!}\right) \xi_i(c), \tag{3.22}$$

and for states in which $s > c$ jobs are in the system we have

$$\xi_i(s) = \xi_i(c)\rho^{s-c}. \tag{3.23}$$

Then using Equations (3.22), (3.23), and the fact that probabilities sum to one, we define

$$\xi_i(c) = \frac{1}{\sum_{s=i}^{c-1}\left[\left(\frac{\mu}{\lambda_H}\right)^{c-s}\left(\frac{c!}{(c-s)!}\right)\right] + \frac{1}{1-\rho}} \tag{3.24}$$

to be the steady-state probability that the system is in state c, given a threshold of $\bar{\imath}^* = i$. Together, Equations (3.22)–(3.24) completely define the stationary distribution of the system, just as one would for an $M/M/c$ queue.

Given these relationships, we define $\overline{R}_i$, the long-run average rate at which a policy with threshold $\bar{\imath}^* = i$ processes type-L jobs, as follows:

$$\overline{R}_i = c\mu - \lambda_H - \left(\sum_{s=i}^{c-1}(c-s)\xi_i(s)\right)\mu. \tag{3.25}$$

The first term of the right-hand side equals the expected number of jobs processed per epoch by all c CSRs. The second term equals the expected number of type-H calls served per period. The last term equals average idle capacity per period—expressed in terms of processing rate per period. The net difference between the first and last two terms is the rate at which type-L jobs are served.

Similarly, we define $\overline{D}_i$, the expected backlog cost, to be

$$\overline{D}_i = \sum_{s=c}^{\infty} \xi_i(s)\, d(s-c) = \xi_i(c)\tilde{d}(\rho). \tag{3.26}$$

Again, the second equality follows from Equation (3.23) and the definition of $\tilde{d}$ in Assumption 1.

Note that as $\bar{\imath}^* = i$ increases, strictly positive terms are removed from the denominator of Equation (3.24), so $\xi_i(c)$ is strictly increasing in i. Further analysis then implies Lemma 6.

LEMMA 6. *Suppose* $\rho < 1$. *Then,*

(i) $0 \equiv \overline{R}_0 < \overline{R}_1 < \cdots < \overline{R}_c \equiv c\mu - \lambda_H$;

(ii) $0 < \overline{D}_0 < \overline{D}_1 < \cdots < \overline{D}_c \equiv (1-\rho)\tilde{d}(\rho)$.

PROOF. Please see the appendix. □

Thus, a given threshold i obtains a pair $(\overline{R}_i, \overline{D}_i)$ that is independent of δ, and as i increases both the intercept and the (negative of the) slope of the line $\overline{R}_i - \delta\overline{D}_i$ increase as well. Furthermore, when $\rho < 1$ both $\overline{R}_i$ and $\overline{D}_i$ are finite for all feasible i, and we can define

$$\delta_i \stackrel{\text{def}}{=} \frac{\overline{R}_i - \overline{R}_{i-1}}{\overline{D}_i - \overline{D}_{i-1}}, \tag{3.27}$$

to obtain the following corollary to Lemma 6. For a graphical view of the corollary's statement, see Figure 1.

COROLLARY 1. *Suppose* $\rho < 1$. *Then* δ_i *is decreasing in* i. *Furthermore,*

(i) *for* $\delta \in [0, \delta_c]$ *a policy with threshold of* $\bar{\imath}^* = c$ *is* δ*-optimal*;

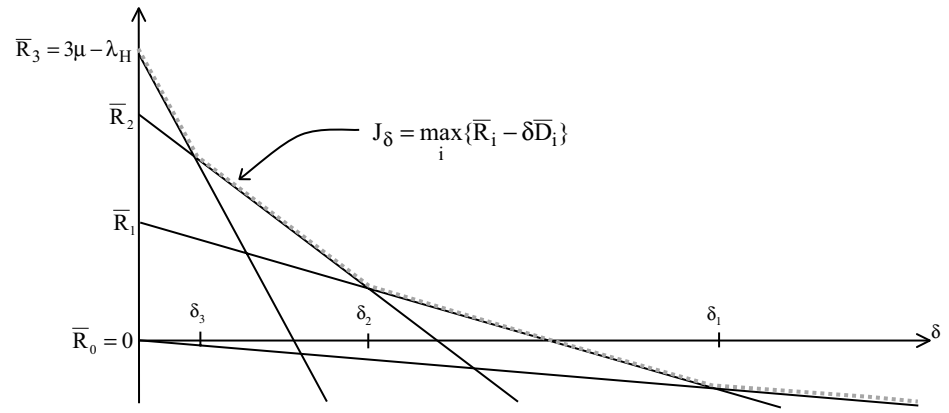
(ii) *for* $\delta \in [\delta_i, \delta_{i-1}]$, $i = 2, \ldots, c$ *a policy with threshold of* $\bar{\imath}^* = i-1$ *is* δ*-optimal*; *and*

(iii) *for* $\delta \in [\delta_1, \infty)$ *a policy with threshold of* $\bar{\imath}^* = 0$ *is* δ*-optimal.*

Lemma 6 and Corollary 1 establish the fundamental relationships that allows us to completely solve the COP. If there exists an i such that $\overline{D}_i = D^*$ then a deterministic policy that gives priority to type-H jobs and has a simple threshold reservation level of $\bar{\imath}^* = i$ for type-L jobs is constrained optimal. Because the set of thresholds is finite, however, it is almost surely the case that $\overline{D}_i \neq D^*$ for all i, and we must randomize between two threshold levels to solve the COP.

Note that when $\overline{D}_i < D^* < \overline{D}_{i+1}$ for some i, reservation thresholds of both i and $i+1$ are δ-optimal at the breakpoint δ_{i+1}. Furthermore, once the system enters a state $s \geqslant i$, the only difference between the two policies is that when i CSRs are busy, the $\bar{\imath}^* = i+1$ policy puts an additional type-L job into service, while the $\bar{\imath}^* = i$ policy does nothing. In this case, by appropriately randomizing between the two thresholds, we can construct a policy π that is δ-optimal for $\delta = \delta_{i+1}$ and has $\overline{D}_\pi = D^*$.

Figure 1. Example of δ-optimal policies when $c = 3$.



DEFINITION 7. A *randomized threshold reservation policy* with *threshold* i^* and *probability* p^* acts as follows at each (before action) event epoch in which there are not type-H calls waiting to be served: (i) if there are i^*+1 or more busy CSRs, then the policy does nothing; (ii) if there are i^* or fewer busy CSRs, then with probability $1-p^*$ the policy puts enough type-L jobs into service so that i^* jobs are in service after action, and with probability p^* the policy puts enough type-L jobs into service so that i^*+1 jobs are in service after action.

THEOREM 1. *Suppose* $\rho<1$. *Then one of four cases exists*:

(i) *if* $\overline{D}_0 > D^*$ *then the COP is infeasible*;

(ii) *if* $\overline{D}_i = D^*$ *for some* $i=0,\dots,c$, *then a simple threshold reservation policy with threshold* $\bar{\imath}^* = i$ *is constrained optimal*;

(iii) *if* $\overline{D}_c < D^*$ *then a simple threshold reservation policy with threshold* $\bar{\imath}^* = c$ *is constrained optimal*; *or*

(iv) *if* $\overline{D}_i < D^* < \overline{D}_{i+1}$ *for some* $i=0,\dots,c-1$, *then a randomized threshold reservation policy which threshold* $i^* = i$ *and probability*

$$p^* = \left(\frac{\lambda_H}{\mu}\right)^{c-i}\left(\frac{(c-i)!}{c!}\right)\left(\sum_{s=i}^{c-1}\left[\left(\frac{\mu}{\lambda_H}\right)^{c-s}\left(\frac{c!}{(c-s)!}\right)\right] + \frac{1}{1-\rho} - \frac{\tilde{d}(\rho)}{D^*}\right) \quad (3.28)$$

is constrained optimal.

PROOF. Parts (i)–(iii) of the theorem follow directly from Lemmas 5 and 6 and Corollary 1.

To calculate the probability in part (iv) one proceeds as follows. First, note that for any $p\in(0,1)$ the randomized routing drives transitions from state $i+1$ to i to be exponentially distributed with mean $((i+1)\mu(1-p))^{-1}$. In turn, the local balance equations imply that $\xi_i(i) = \big((i+1)\mu(1-p)/\lambda_H\big)\xi_i(i+1)$, so that Equation (3.24) becomes

$$\xi_{i,p}(c) \equiv \left\{\left(\frac{\mu}{\lambda_H}\right)^{c-i}\frac{c!}{(c-i)!}(1-p) + \sum_{s=i+1}^{c-1}\left[\left(\frac{\mu}{\lambda_H}\right)^{c-s}\frac{c!}{(c-s)!}\right] + \frac{1}{1-\rho}\right\}^{-1}. \quad (3.29)$$

Observe that $\xi_{i,0}(c)\tilde{d}(\rho) < D^*$, $\xi_{i,1}(c)\tilde{d}(\rho) > D^*$, and $\xi_{i,p}(c)$ is increasing in p. Thus, we can set $\xi_{i,p}(c)\tilde{d}(\rho) = D^*$ and solve for p to complete the proof. □

The determination of the optimal policy requires about the computational effort needed to calculate the normalizing constant of an $M/M/c$ queue (Equation (3.24) with $i=0$). To make the calculation explicit, we first transform the problem as follows.

To find $\bar{\imath} = \max\{i \mid \xi_i(c)\tilde{d}(\rho) \leqslant D^*\}$, we substitute for $\xi_i(c)$ using Equation (3.24) and rearrange terms as follows:

$$\bar{\imath} = \max\left\{i \,\middle|\, \sum_{s=i}^{c-1}\frac{(\mu/\lambda_H)^{c-s}}{(c-s)!} \geqslant \frac{1}{c!}\left(\frac{\tilde{d}(\rho)}{D^*} - \frac{1}{1-\rho}\right)\right\} = \min\left\{i \,\middle|\, \sum_{s=1}^{i}\frac{(\mu/\lambda_H)^{s}}{s!} \geqslant \frac{\tilde{\mathscr{D}}(\rho)}{c!}\right\}, \quad (3.30)$$

where

$$\tilde{\mathscr{D}}(\rho) \stackrel{\text{def}}{=} \sum_{q=0}^{\infty}\rho^q\left(\frac{d(q)-D^*}{D^*}\right) = \sum_{q=0}^{\infty}\rho^q\frac{d(q)}{D^*} - \sum_{q=0}^{\infty}\rho^q = \frac{\tilde{d}(\rho)}{D^*} - \frac{1}{1-\rho} \quad (3.31)$$

is a shifted and scaled version of the expected backlog cost.

Then to find the optimal policy parameters, one first calculates the right-hand side of the inequality inside the minimization of Equation (3.30). If it is less than or equal to zero, then a simple threshold of $\bar{\imath}^* = c$ is optimal. Otherwise, one adds terms to the left-hand side one at a time until the left-hand side satisfies the inequality. If no $\bar{\imath}$ satisfies the inequality, then the COP is infeasible. If for some $\bar{\imath}\in\{1,\dots,c-1\}$ the left-hand side satisfies the inequality exactly (with equality), then a simple threshold of $\bar{\imath}^* = \bar{\imath}$ is optimal. Otherwise, a randomized threshold policy with $i^* = \bar{\imath}-1$ and

$$p^* = 1 - \left(\frac{\tilde{\mathscr{D}}(\rho)}{c!} - \sum_{s=1}^{i^*}\frac{(\mu/\lambda_H)^s}{s!}\right)\frac{(i^*+1)!}{(\mu/\lambda_H)^{i^*+1}} \quad (3.32)$$

is optimal.

It is also interesting to note that if we multiply both sides of the inequality inside the minimization of Equation (3.30) by $e^{-\mu/\lambda_h}$, then the left-hand-side becomes the probability that a sample of a Poisson distribution of mean μ/λ_H falls between 1 and $\bar{\imath}$.

3.5. Service-Level Constraints Based on Delay

The service-level constraints developed in this section, although quite general, are formulated as functions of the number in queue. Typically, however, service-level constraints are formulated as a function of the delay (time) in queue. We therefore briefly demonstrate how the two most common examples of both types of constraints can be formulated as functions of $\xi_i(c)$, the steady-state probability that all c servers are busy.

In call centers, service-level constraints based on the average delay in queue are commonly called constraints on the average speed of answer (ASA). Typically, the α for ASA is set at 20 or 30 seconds. We can use Little's law to transform an ASA-based constraint to an occupancy-based constraint as follows.

Let Q denote the number in queue in steady state so that $ASA \leqslant \alpha \Leftrightarrow \mathsf{E}[Q] \leqslant \alpha/\lambda_H$. Then using $d(\bar{q}) \stackrel{\text{def}}{=} \bar{q}$ so that

$\tilde{d}(\rho) = \sum_{q=0}^{\infty} q\rho^q = \rho/(1-\rho)^2$, we have

$$\overline{D}_i \stackrel{\text{def}}{=} \xi_i(c)\frac{\rho}{(1-\rho)^2}, \tag{3.33}$$

and $D^* \stackrel{\text{def}}{=} \alpha/\lambda_H$. Then $\overline{D}_i \leqslant D^*$ if and only if $\xi_i(c) \leqslant (1-\rho)^2\alpha/\rho\lambda_H$.

Call centers may also use a lower-bound constraint, α, on the fraction of customers that are delayed less than β units of time. The most common numbers used for α and β are 20 seconds and 0.8, so that "at least 80% of the customers are served in 20 seconds or less." Mandelbaum (2001) has conjectured that these numbers result from a (mis)application of the "80-20 rule." For another history of how this standard was arrived upon, see Fleischer (2000, item 71).

Given the priority of type-H jobs, we know that when there is a positive queue length the system behaves as a standard $M/M/c$ queue with arrival rate λ_H and service rate μ. For this system it is well known that the conditional distribution of delay of an arriving customer—{delay | delay > 0}—is exponentially distributed with mean $(c\mu - \lambda_H)^{-1}$ (for example, see Wolff 1989, §5.9).

From PASTA and Equation (3.24), we also know that the probability that an arriving call finds all c servers busy and is delayed equals $\sum_{s=c}^{\infty} \xi_i(s) = \xi_i(c)\sum_{q=0}^{\infty}\rho^q = \xi_i(c)/(1-\rho)$. Therefore,

$$\mathbf{P}\{\text{delay} > \beta\} = \frac{\xi_i(c)}{1-\rho}\, e^{-(c\mu-\lambda_H)\beta} \leqslant (1-\alpha) \tag{3.34}$$

if and only if $\xi_i(c) \leqslant (1-\alpha)(1-\rho)\, e^{(c\mu-\lambda_H)\beta}$.

Thus, we can let $d(\bar{q}) \stackrel{\text{def}}{=} \mathbf{1}\{\bar{q} > 0\}$ so that $\tilde{d}(\rho) = \sum_{q=0}^{\infty} d(q)\rho^q = \sum_{q=1}^{\infty}\rho^q = \rho/(1-\rho)$, and $\overline{D}_i = \xi_i(c)\rho/(1-\rho)$. Similarly, we can let $D^* \stackrel{\text{def}}{=} \rho(1-\alpha)e^{(c\mu-\lambda_H)\beta}$. Together, these definitions of $\overline{D}_i$ and D^* mimic the delay-based constraint.

4. DIFFERENT SERVICE-TIME DISTRIBUTIONS: $\mu_H \neq \mu_L$

In this section we assume that type-H and type-L calls have different service-time distributions: $\mu_H \neq \mu_L$. Accordingly, we increase the state space dimension by one so that $S = \{(i, j, q) \mid 0 \leqslant i + j \leqslant c, q \geqslant 0\}$. Here i and j are the numbers of CSRs serving type-H and type-L jobs, and q is the number of type-H jobs in queue. We also uniformize state transitions at rate $\lambda_H + c\mu_H + c\mu_L = 1$.

While the state space expands, the other elements of the model developed in §3 remain the same. In particular, we continue to define the COP as in Equations (3.3)–(3.5), and we continue to assume that the backlog cost behaves as defined in Assumption 1.

The fact that the service time distributions for type-H and type-L jobs differ prevents us from proving that type-H priority policies are optimal, however. When $\mu_H \neq \mu_L$ then the coupling argument used in the first part of the proof of Lemma 1 breaks down.

Still, this class remains attractive. It is both natural and common to give higher priority to the work for which there exists a strict service-level constraint, rather than to the work that is postponable. Furthermore, given the priority of type-H calls, the argument used in Lemma 1 to prove the optimality of type-H work-conserving policies can be used without modification.

LEMMA 7. *Among type-H priority policies, type-H work conserving policies are optimal.*

PROOF. Please see Step 2 in the proof of Lemma 1, in the appendix. □

Then given the class of type-H priority, type-H work-conserving policies, we can again, with some effort, collapse the infinite-dimensional state space of the original problem into one of finite dimension and formulate a finite LP. While the optimal solution to the LP is not guaranteed to be optimal among all policies, it is optimal within the class of type-H priority policies.

Thus, given priority policies, we can again formulate a reduced-dimensional state space and embed the priority of type-H calls into the state transition equations. More specifically, we let $S = \{(i, j) \mid 0 \leqslant i,\ 0 \leqslant j \leqslant c\}$, where i represents the number of type-H jobs in service or in queue, and j represented the number of type-L jobs in service. Note that $\min\{i, c-j\}$ represents the number of type-H jobs in service. Then the state transition equation becomes

$$(i_{t+1}, j_{t+1}) = \begin{cases} (\bar{\imath}_t+1, \bar{\jmath}_t), & \text{w. p. } \lambda_H; \\ (\bar{\imath}_t-1, \bar{\jmath}_t), & \text{w. p. } \min\{\bar{\imath}_t, c-\bar{\jmath}_t\}\cdot\mu_H; \\ (\bar{\imath}_t, \bar{\jmath}_t-1), & \text{w. p. } \bar{\jmath}_t\cdot\mu_L; \text{ and} \\ (\bar{\imath}_t, \bar{\jmath}_t), & \text{w. p. } (c-\min\{\bar{\imath}_t, c-\bar{\jmath}_t\})\cdot\mu_H \\ & \qquad +(c-\bar{\jmath}_t)\cdot\mu_L. \end{cases} \tag{4.1}$$

These transition equations parallel Equation (3.1). Note that when $\bar{\imath}_t = 0$ the transition to $(\bar{\imath}_t - 1, \bar{\jmath}_t)$ occurs with probability zero, and when $\bar{\jmath}_t = 0$ the transition to $(\bar{\imath}_t, \bar{\jmath}_t - 1)$ occurs with probability zero. In addition, the transition in which $(i_{t+1}, j_{t+1}) = (\bar{\imath}_t, \bar{\jmath}_t)$ is the result of uniformization at rate $\lambda_H + c\mu_H + c\mu_L = 1$.

To formulate the LP when $\mu_H \neq \mu_L$ we proceed as before. First, we formulate the infinite-dimensional LP that explicitly represents states in which there exists a positive queue length and relaxes the service-level constraint using a Lagrange multiplier. Then, we characterize the distribution of the number of type-H jobs in queue as a function of the states in which all servers are busy and no jobs are queued. Note that, because $\mu_H \neq \mu_L$, there are now $(c+1)$ such states in which $i + j = c$, and the characterization is based on a linear combination of $(c+1)$ geometric series. Finally, given this characterization, we reduce the infinite-dimensional LP to a finite-dimensional formulation and reintroduce the service-level constraint.

4.1. The Infinite-Dimensional LP When $\mu_H \neq \mu_L$

We can extend the arguments of Lemma 2 to prove analogous results when $\mu_H \neq \mu_L$. More specifically, we again let

the vector ξ_π be the solution to $\xi = \xi M_\pi$, $\xi \mathbf{1} = 1$ so that, for any stationary policy $\pi \in \Pi$, $\xi_\pi(s,a)$ is the stationary probability of being in state s and taking action a. Then we have Lemma 8.

LEMMA 8. *Suppose $\rho = \lambda_H/(c\mu_H) < 1$. Then among type-H priority policies, there exist stationary, deterministic policies that are δ-optimal. Furthermore, each stationary deterministic policy π:*

(i) *induces a single, positive recurrent class of states, and the expected absorption time into that class is finite*;

(ii) *has limiting state-action frequencies which correspond to the stationary distribution of the induced Markov chain*: $\lim_{n\to\infty} \frac{1}{n}\sum_{t=0}^{n-1} \mathbf{1}\{(s_t,a_t)=(s,a)\} = \xi_\pi(s,a)$ *w.p.* 1 $\forall\ s_0$; *and*

(iii) *has uniformly integrable one-period revenues*:

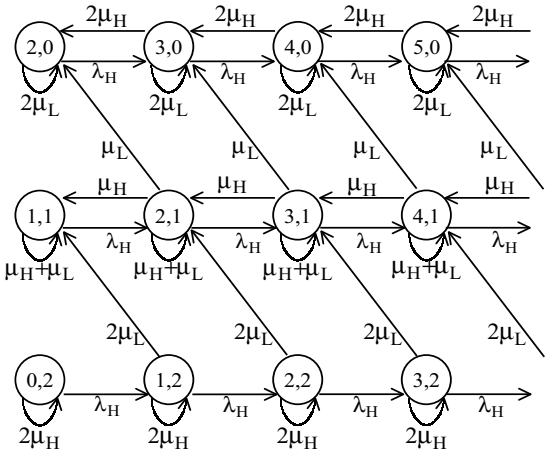
$$\sum_{s\in S, a\in A_s} \xi_\pi(s,a)|R(s,a) - \delta D(s,a)| < \infty.$$

PROOF. Please see the appendix. □

Again, parts (i)–(iii) of the lemma allow us to apply of Altman and Schwartz's (1991) Theorem 7.1 to demonstrate that there is a one-to-one correspondence between stationary policies and feasible solutions to an appropriately defined infinite-dimensional LP. The fact that stationary policies are optimal allows us to conclude that an optimal solution to an infinite-dimensional LP finds a δ-optimal policy.

As in §3, we split into two sets the LP constraints that define the system balance equations. For states in which $i+j<c$, we let $K_{(i,j)} \stackrel{\text{def}}{=} \{(i',j',a)\,|\,(i',j')\in S,\ a\in A_{(i',j')},\ i=1,\ j'+a=j\}$ be the set of state-action pairs (i',j',a) that takes the system from state (i',j') to state (i,j) by admitting a type-L jobs. For states in which $i+j\geqslant c$, the priority of type-H jobs implies that there are no decisions to be made, and the state transition equations (4.1) completely determine the behavior of the system. Figure 2 shows the relevant portion of the corresponding state transition diagram for a system with $c=2$ servers.

Figure 2. Transition diagram for states $i+j>c$ with $c=2$.



Then we have

$$\max \sum_{i=0}^{c-1}\sum_{j=0}^{c-i-1}\sum_{a=0}^{c-i-j} a\xi(i,j,a) - \delta\sum_{j=0}^{c}\sum_{i=c-j+1}^{\infty} d(i+j-c)\xi(i,j,0), \tag{4.2}$$

$$\begin{aligned}\text{s.t.} \sum_{a=0}^{(c-i-j)^+} \xi(i,j,a) &= \lambda_H \sum_{(i',j',a)\in K_{(i-1,j)}} \xi(i',j',a) \\ &\quad + \min\{c-j, i+1\}\mu_H \sum_{(i',j',a)\in K_{(i+1,j)}} \xi(i',j',a) \\ &\quad + (j+1)\mu_L \sum_{(i',j',a)\in K_{(i,j+1)}} \xi(i',j',a) \\ &\quad + (\max\{c-i, j\}\mu_H + (c-j)\mu_L) \\ &\quad \times \sum_{(i',j',a)\in K_{(i,j)}} \xi(i',j',a) \quad \forall\, 0\leqslant i+j\leqslant c+1,\end{aligned} \tag{4.3}$$

$$\begin{aligned}\xi(i,j,0) = \lambda_H\xi(i-1,j,0) &+ (c-j)\mu_H\xi(i+1,j,0) \\ &+ (j+1)\mu_L\xi(i,j+1,0) \\ &+ (j\mu_H + (c-j)\mu_L)\xi(i,j,0) \\ &\qquad \forall\, j\leqslant c-1,\ i+j\geqslant c+2,\end{aligned} \tag{4.4}$$

$$\xi(i,j,0) = \lambda_H\xi(i-1,j,0) + c\mu_H\xi(i,j,0) \qquad \forall\, j=c,\ i\geqslant 2, \tag{4.5}$$

$$\sum_{i=0}^{\infty}\sum_{j=0}^{c}\sum_{a=0}^{(c-i-j)^+} \xi(i,j,a) = 1, \tag{4.6}$$

$$\xi(i,j,a)\geqslant 0 \quad \forall\ i,j,a\in A_{(i,j)}. \tag{4.7}$$

Again, whenever $K_i = \varnothing$ we define the associated summation to equal zero.

The formulation is a direct analogue to the infinite dimensional LP when $\mu_H = \mu_L$. The objective function (4.2), the constraint that probabilities sum to one (4.6), and the nonnegativity constraints (4.7) are straightforward. Balance constraints when $i+j\leqslant c+1$ (Equation (4.3)) are also direct analogues. There are now two sets of balance constraints for the tail of the queue, however. For states in which $j<c$ and there is at least one type-H job in service, Equation (4.4) holds; and for states in which $j=c$ and there are no type-H jobs in service, Equation (4.5) applies. Note that the last term on the right-hand side of all balance constraints (4.3)–(4.5) is flow into a state due to uniformization.

4.2. The Number in Queue When $\mu_H \neq \mu_L$

To reduce the infinite dimensional problem (4.2)–(4.7) to a finite LP we must find a finite form for the infinite sets of

balance constraints (4.4)–(4.5) and, as before, we can find an appropriate set of geometric series to substitute for these equations. Because there are now $(c+1)$ states associated with each queue length, the single geometric series of §3 now becomes a set of $(c+1)$ geometric series.

We begin the analysis by restating the balance equations (4.4)–(4.5) as follows. First, we write the left-hand side of both equations as $(\lambda_H + c\mu_H + c\mu_L)\xi(i,j,0)$. Then, we carry the uniformization terms on the right-hand sides to the left, equivalently restating the equations as they would appear in the standard formulation of a CTMC. Finally, we divide the equations by the remaining coefficients on the left-hand sides so that we have

$$\xi(i,j,0) = p_1(j)\xi(i-1,j,0) + p_2(j)\xi(i+1,j,0) + p_3(j)\xi(i,j+1,0) \quad \forall\, i+j \geqslant c+2, 0 \leqslant j < c, \tag{4.8}$$

$$\xi(i,j,0) = p_1(c)\xi(i-1,c,0) \quad \forall\, i+j \geqslant c+2, j = c, \tag{4.9}$$

where

$$p_1(j) \stackrel{\text{def}}{=} \frac{\lambda_H}{\lambda_H + (c-j)\mu_H + j\mu_L}, \tag{4.10}$$

$$p_2(j) \stackrel{\text{def}}{=} \frac{(c-j)\mu_H}{\lambda_H + (c-j)\mu_H + j\mu_L}, \quad \text{and} \tag{4.11}$$

$$p_3(j) \stackrel{\text{def}}{=} \frac{(j+1)\mu_L}{\lambda_H + (c-j)\mu_H + j\mu_L}. \tag{4.12}$$

Next, for each j we consider the quadratic equation

$$g(j,z) \stackrel{\text{def}}{=} p_2(j)z_j^2 - z_j + p_1(j) = 0 \tag{4.13}$$

with roots $z_j \leqslant z_j'$. These roots will become the rates at which the geometric series converge (or diverge) as the size of the backlog grows large. From elementary considerations we can derive the following properties of the roots of the $c+1$ equations.

LEMMA 9. (i) $z_0 = \lambda_H/(c\mu_H) = \rho < 1$ *and* $z_0' = 1$;
(ii) $0 < z_j < 1 < z_j'$ *for* $0 < j < c$;
(iii) $z_c = \lambda_H/(\lambda_H + c\mu_L) < 1$ *and* z_c' *is not defined.*

PROOF. Please see the appendix. □

It can be shown that, as long as $\rho < 1$ so that the system is stable, the generating functions associated with $c+1$ series $\{\xi(i,j,0) \mid i \geqslant c-j+1\}$ for $j = 0,\ldots,c$-are well defined in terms of the smaller roots and can be used to derive the steady state probabilities. The following lemma summarizes the results.

LEMMA 10. *Suppose* $\rho < 1$. *Then one of two cases exists.*

(i) *If* $\mu_H \leqslant \mu_L$ *or* $\lambda_H \neq c(\mu_H - \mu_L)$, *then* $z_i \neq z_j$ *for* $i \neq j$, *and for any* $0 \leqslant j \leqslant c$,

$$\xi(c-j+q,j,0) = \sum_{k=j}^{c} a_{j,k} z_k^q, \quad \forall\, q \geqslant 1, \tag{4.14}$$

where

$$a_{c,c} = \sum_{a=0}^{c} \xi(0, c-a, a), \tag{4.15}$$

$$a_{j,j} = \sum_{a=0}^{j} \xi(c-j, j-a, a) + \frac{p_3(j)}{p_2(j)} \sum_{a=0}^{j+1} \xi(c-j-1, j+1-a, a) + \frac{p_3(j)z_j}{p_2(j)(z_j'-z_j)} \sum_{l \geqslant j+1} \frac{a_{j+1,l}}{1-z_l/z_j} - \frac{p_3(j)z_j'}{p_2(j)(z_j'-z_j)} \sum_{l \geqslant j+1} \frac{a_{j+1,l}}{1-z_l/z_j'} \quad j = 0,\ldots,c-1, \tag{4.16}$$

$$a_{j,k} = \frac{-p_3(j)}{p_2(j)(1-z_j/z_k)(1-z_j'/z_k)} a_{j+1,k}. \quad \forall\, 0 \leqslant j < k \leqslant c. \tag{4.17}$$

(ii) *If* $\mu_H > \mu_L$ *and* $\lambda_H = c(\mu_H - \mu_L)$, *then* $z_j = 1 - \mu_L/\mu_H, \forall j$ *and* $z_j' = c/(c-j), \forall j < c$. *Letting* $z^* = 1 - \mu_L/\mu_H$, *we have*

$$\xi(c-j+q,j,0) = \sum_{k=1}^{c-j+1} a_{j,k+j-1} \binom{q+k-1}{k-1} z^{*q}, \quad \forall\, q \geqslant 1, \tag{4.18}$$

where $a_{j,k}$ *can be computed by taking the* $(c-j+1-k)$th *derivative of* $f_j(z)(1-z^*z)^{c-j+1}$ *and evaluating at* $1/z^*$.

PROOF. Please see the appendix. □

REMARK 2. To prove Lemma 10, we extend the balance Equations (4.8) and (4.9) to include $q = 1$. This implies that Equations (4.14) and (4.18) hold for $q = 0$ as well. For details, please see the proof of Lemma 10 in the appendix.

Note that the summations $\sum_a \xi(\cdot, \cdot - a, a)$ in Equations (4.15) and (4.16) represent the steady-state probabilities of being in boundary states $\{(c-j,j) \mid 0 \leqslant j \leqslant c\}$ after action. Thus, Lemma 10 allows us to express the tails of the steady-state distribution of the queue as linear combinations of the after-action probabilities of being in these boundary states, and it enables us to reduce the infinite LP representation of the MDP to one with a finite number of states and constraints. We note that the conditions assumed in part (i) of the lemma occur almost surely, and in the remainder of the paper we focus on this case. Similar results can be derived for case (ii).

More generally, observe that for any fixed j, the tail distribution of the number-in-system can be described as the linear combination of $c-j+1$ geometric series with rates $z_j,\ldots,z_c$. Indeed, the flows detailed in Figure 2 clearly reflect this recursive structure. For example, for the states on the bottom row, in which $j = c$ type-L and no type-H jobs are in service, the rate of

decay of the tail, $\xi(i+1,c,0)=\xi(i,c,0)z_c$ or equivalently $\xi(i+1,c,0)=\xi(i,c,0)(\lambda_H/(\lambda_H+c\mu_L))$, reflects this structure. Next, flows into states in which $j=c-1$ type-L and one type-H jobs are in service have flows in from states within the same row and from states in the row below (where $j=c$). Therefore, the tail decays with a combination of rates z_c and z_{c-1}. As we move from j to $j-1$ type-L jobs in service, we add another series, z_{j-1}. Inductively, this holds true for all rows.

This structure clearly holds whenever such a system is operated under a priority policy that is work conserving for the high-priority class. The resulting closed-form expressions—and the calculations that lead to them—are likely to be useful in contexts beyond the specific system analyzed in this paper.

4.3. The Finite-Dimensional LP When $\mu_H \neq \mu_L$

When $\rho<1$ the conditions of case (i) of Lemma 10 hold almost surely. In this case, the reduction to a finite dimensional LP is straightforward.

We first (numerically) find the roots of $p_1(j)$, $p_2(j)$, and $p_3(j)$ for $0\leqslant j\leqslant c$. Then, for each $0\leqslant j\leqslant c$ and $i+j\geqslant c$ we can use Equation (4.14) to substitute for the $\xi(i,j,0)$s. This allows us, in turn, to develop closed-form expressions for the terms involving the tail probabilities in Equations (4.2) and (4.6). More specifically, for the left-hand side of Equation (4.6), we have

$$\begin{aligned}
&\sum_{i=0}^{\infty}\sum_{j=0}^{c}\sum_{a=0}^{(c-i-j)^+}\xi(i,j,a)\\
&\quad=\sum_{i=0}^{c}\sum_{j=0}^{c-i}\sum_{a=0}^{c-i-j}\xi(i,j,a)+\sum_{j=0}^{c}\sum_{i=c-j+1}^{\infty}\xi(i,j,0)\\
&\quad=\sum_{i=0}^{c}\sum_{j=0}^{c-i}\sum_{a=0}^{c-i-j}\xi(i,j,a)+\sum_{j=0}^{c}\sum_{i=c-j+1}^{\infty}\sum_{k=j}^{c}a_{j,k}z_k^{i+j-c}\\
&\quad=\sum_{i=0}^{c}\sum_{j=0}^{c-i}\sum_{a=0}^{c-i-j}\xi(i,j,a)+\sum_{j=0}^{c}\sum_{k=j}^{c}a_{j,k}\sum_{i=c-j+1}^{\infty}z_k^{i+j-c}\\
&\quad=\sum_{i=0}^{c}\sum_{j=0}^{c-i}\sum_{a=0}^{c-i-j}\xi(i,j,a)+\sum_{j=0}^{c}\sum_{k=j}^{c}a_{j,k}\frac{z_k}{1-z_k}. \qquad (4.19)
\end{aligned}$$

Similarly, for the second term in the objective function (4.2), we have

$$\begin{aligned}
&\delta\sum_{j=0}^{c}\sum_{i=c-j+1}^{\infty}d(i+j-c)\xi(i,j,0)\\
&\quad=\delta\sum_{j=0}^{c}\sum_{i=c-j+1}^{\infty}d(i+j-c)\left(\sum_{k=j}^{c}a_{j,k}z_k^{i+j-c}\right)\\
&\quad=\delta\sum_{j=0}^{c}\sum_{k=j}^{c}a_{j,k}\sum_{i=c-j+1}^{\infty}z_k^{i+j-c}d(i+j-c)\\
&\quad=\delta\sum_{j=0}^{c}\sum_{k=j}^{c}a_{j,k}\tilde{d}(z_k). \qquad (4.20)
\end{aligned}$$

Finally, we use Equations (4.14)–(4.17) to substitute for the infinite sets of constraints (4.4) and (4.5), as well as the $2(c+1)$ constraints (4.3) for which $i+j$ equals c or $c+1$. For the $c+1$ states in which $i+j=c+1$, Equations (4.15)–(4.17) can be directly substituted to eliminate the constraints (4.3). For states in which $i+j=c$, we use Equation (4.14) to modify the constraints Equation (4.3) as follows:

$$\begin{aligned}
&\xi(c-j,j,0)\\
&\quad=\lambda_H\sum_{(i',j',a)\in K_{(c-j-1,j)}}\xi(i',j',a)\\
&\qquad+(c-j)\mu_H\xi(c-j+1,j,0)\\
&\qquad+(j+1)\mu_L\xi(c-j,j+1,0)\\
&\qquad+(j\mu_H+(c-j)\mu_L)\sum_{(i',j',a)\in K_{(c-j,j)}}\xi(i',j',a)\\
&\quad=\lambda_H\sum_{(i',j',a)\in K_{(c-j-1,j)}}\xi(i',j',a)+(c-j)\mu_H\sum_{k=j}^{c}a_{j,k}z_k\\
&\qquad+(j+1)\mu_L\sum_{k=j+1}^{c}a_{j+1,k}z_k\\
&\qquad+(j\mu_H+(c-j)\mu_L)\sum_{(i',j',a)\in K_{(c-j,j)}}\xi(i',j',a). \qquad (4.21)
\end{aligned}$$

Thus, with the addition of the $(c+1)(c+2)/2$ variables $\{a_{j,k}\,|\,0\leqslant j\leqslant k\leqslant c\}$ we can reduce the infinite LP to a finite one, and for any fixed $\delta\in[0,\infty)$, the optimal solution to the reformulated LP will find a type-H priority policy that is δ-optimal.

In turn, if we eliminate the Lagrangian term (4.20) from the objective function and, instead, reintroduce the constraint

$$\sum_{j=0}^{c}\sum_{k=j}^{c}a_{j,k}\tilde{d}(z_k)\leqslant D^*, \qquad (4.22)$$

then the optimal solution to the LP will generate a dual price δ^* for the new constraint. Furthermore, the optimal solution of the LP will be δ-optimal for δ^*. Thus, from Lemma 5 we know that, among type-H priority policies, the following LP finds a constrained optimal policy:

$$\max\ \sum_{\substack{i=0\\ c-i-j}}^{c-1}\sum_{j=0}^{c-i-1}\sum_{a=0}^{c-i-j}a\xi(i,j,a), \qquad (4.23)$$

$$\begin{aligned}
\text{s.t.}\ &\sum_{a=0}^{}\xi(i,j,a)\\
&\quad=\lambda_H\sum_{(i',j',a)\in K_{(i-1,j)}}\xi(i',j',a)\\
&\qquad+(i+1)\mu_H\sum_{(i',j',a)\in K_{(i+1,j)}}\xi(i',j',a)\\
&\qquad+(j+1)\mu_L\sum_{(i',j',a)\in K_{(i,j+1)}}\xi(i',j',a)\\
&\qquad+((c-i)\mu_H+(c-j)\mu_L)\sum_{(i',j',a)\in K_{(i,j)}}\xi(i',j',a)\\
&\qquad\qquad 0\leqslant i+j\leqslant c-1, \qquad (4.24)
\end{aligned}$$

$$\xi(c-j,j,0)=\lambda_H \sum_{(i',j',a)\in K_{(c-j-1,j)}} \xi(i',j',a) + (c-j)\mu_H \sum_{k=j}^{c} a_{j,k} z_k + (j+1)\mu_L \sum_{k=j+1}^{c} a_{j+1,k} z_k + (j\mu_H + (c-j)\mu_L) \sum_{(i',j',a)\in K_{(c-j,j)}} \xi(i',j',a) \qquad 0 \leqslant j \leqslant c-1, \quad (4.25)$$

$$a_{c,c}=\sum_{a=0}^{c} \xi(0,c-a,a), \quad (4.26)$$

$$a_{j,j}=\sum_{a=0}^{j} \xi(c-j,j-a,a) + \frac{p_3(j)}{p_2(j)} \sum_{a=0}^{j+1} \xi(c-j-1,j+1-a,a) + \frac{p_3(j) z_j}{p_2(j)(z'_j - z_j)} \sum_{l \geqslant j+1} \frac{a_{j+1,l}}{1-z_l/z_j} - \frac{p_3(j) z'_j}{p_2(j)(z'_j - z_j)} \sum_{l \geqslant j+1} \frac{a_{j+1,l}}{1-z_l/z'_j} \qquad \forall\ 0 \leqslant j \leqslant c-1, \quad (4.27)$$

$$a_{j,k}=\frac{-p_3(j)}{p_2(j)(1-z_j/z_k)(1-z'_j/z_k)} a_{j+1,k} \qquad \forall\ 0 \leqslant j < k \leqslant c, \quad (4.28)$$

$$\sum_{j=0}^{c} \sum_{k=j}^{c} a_{j,k} \tilde{d}(z_k) \leqslant D^*, \quad (4.29)$$

$$\sum_{i=0}^{c} \sum_{j=0}^{c-i} \sum_{a=0}^{c-i-j} \xi(i,j,a) + \sum_{j=0}^{c} \sum_{k=j}^{c} a_{j,k} \frac{z_k}{1-z_k} = 1, \quad (4.30)$$

$$\xi(i,j,a) \geqslant 0 \quad \forall\ 0 \leqslant i+j+a \leqslant c. \quad (4.31)$$

Here, the objective function (4.23) drops the Lagrangian term (4.20). The $c(c+1)/2$ constraints (4.24) correspond exactly to those in Equation (4.3) for which $i+j \leqslant c-1$. The c constraints (4.25) correspond to (4.21), and the $(c+1)(c+2)/2$ constraints (4.26)–(4.28) define the $a_{j,k}$ terms as in (4.15)–(4.17). The service-level constraint (4.29) follows from (4.22), and the constraint (4.31) that probabilities sum to one follows from (4.19). Again, the LP formulation drops one redundant balance constraint, (4.25) for $j=c$. (See Puterman 1994, §8.8).

This proves our main result for cases in which $\mu_H \neq \mu_L$.

THEOREM 2. *Suppose that $\rho < 1$ and that either $\mu_H \leqslant \mu_L$ or $\lambda_H \neq c(\mu_H - \mu_L)$. If the LP (4.23)–(4.31) is feasible, then its optimal solution finds a policy that is constrained optimal among all type-H priority policies.*

If the LP is feasible and the dual price of (4.29) $\delta^* > 0$, then the optimal solution is δ-optimal for δ^*. Here, δ^* is the extra throughput of type-L calls that can be achieved per unit that the service-level constraint D^* is relaxed. If the dual price of (4.29) $\delta^* = 0$, however, then the service-level constraint is not tight, and it is optimal to put a type-L job into service whenever a server becomes idle and there are no type-H calls in queue.

If the LP (4.23)–(4.31) is not feasible, then there exists no feasible type-H priority policy. In particular, a type-H work-conserving policy that *never* puts type-L jobs into service is not feasible. This implies that when the LP is not feasible, the COP itself has no feasible solution.

Note that the LP (4.23)–(4.31) has $(c+1)(c+2)(c+6)/6+1$ variables: $(c+1)(c+2)(c+3)/6$ of the variables are the $\xi(i,j,a)$s, $(c+1)(c+2)/2$ are the $a_{j,k}$s, and one is the slack variable in (4.29). Similarly, adding up the constraints (4.24)–(4.30) we find there are $c(c+3)$ constraints.

We can reduce the size of the LP, however, by eliminating the variables $\{a_{j,k} | j \neq k\}$. To do this we first substitute the right-hand sides of Equation (4.28) for the appropriate $a_{j,k}$s found in the constraints (4.25)–(4.27). In turn, this allows us to eliminate the variables $\{a_{j,k} | j \neq k\}$ and the constraints (4.28). The result of this substitution reduces the LP to one with $(c+1)(c+2)(c+4)/6+1$ variables and $c(c+5)/2$ constraints. (Similar substitution would allow us to eliminate the $a_{j,j}$s as well.)

4.4. Service-Level Constraints Based on Delay

Again, we can use Little's law to formulate occupancy based equivalents of constraints based on ASA, the average delay in queue. Suppose the upper limit on expected delay is α. Then we can define $d(\bar{q}) = \bar{q}$ so that

$$\sum_{j=0}^{c} \sum_{i=c-j+1}^{\infty} \sum_{k=j}^{c} a_{j,k} z_k^{i+j-c} d(i+j-c) = \sum_{j=0}^{c} \sum_{k=j}^{c} a_{j,k} \sum_{i=c-j+1}^{\infty} z_k^{i+j-c} (i+j-c) = \sum_{j=0}^{c} \sum_{k=j}^{c} a_{j,k} \frac{z_k}{(1-z_k)^2} \leqslant \lambda_H \alpha$$

is the appropriate service-level constraint.

When $\mu_H \neq \mu_L$ the distribution of delay becomes more difficult to characterize. One such constraint that can be defined exactly in a straightforward fashion, however, is $\mathbf{P}\{\text{delay} > 0\}$. In particular, from PASTA we have $\mathbf{P}\{\text{delay} > 0\} \Leftrightarrow \mathbf{P}\{i+j \geqslant c\}$. Then if α is an upper bound on the probability of delay upon arrival, we can define

$$1 - \sum_{i+j+a<c} \xi(i,j,a) \leqslant \alpha$$

as the service-level constraint.

5. DISCUSSION

For cases in which $\mu_H = \mu_L$ the results are fairly complete. We have been able to demonstrate the optimality of randomized threshold reservation policies and to reduce the

problem of finding the optimal policy parameters, i^* and p^*, to computation that grows (pseudo) linearly with the number of servers.

Together, the small number of parameters and the ease of their computation make these policies attractive to implement in call centers. In these environments, estimated arrival rates and staffing levels change every 15 or 30 minutes. Even for very large call centers, new policy parameters can easily be calculated and implemented within this timeframe.

At the same time, limiting results for the case of $\mu_H = \mu_L$ would be of further theoretical interest and would allow for closed-form expressions for (approximately) optimal threshold values. We note that §3.4's characterization of the optimal (deterministic) threshold as the inverse of the cumulative distribution of a Poisson random variable may be of use in this respect.

For the case in which $\mu_H \neq \mu_L$, we view our results as promising. The optimal solution to the LP can be used to calculate both the optimal actions and the steady-state distribution of the induced Markov chain. (See Puterman 1994, §8.8.) Although we have proved only the optimality of LP (4.23)–(4.30) among type-H priority policies, we believe that they should perform quite well on an absolute basis.

In this case, the computational effort required to calculate optimal policy parameters grows substantially, however. For example, for a pool of 100 CSRs, a reduced version of the LP, which eliminates the $a_{j,k}$s, requires roughly 176,850 variables and 5,150 constraints. Furthermore, the LP's optimal solution describes a routing policy with 5,250 actions, one for each state of the MDP.

Although this size of problem is well within the capabilites of current LP solvers, for large call centers these types of policies may be difficult to implement. Therefore, additional work is warranted to develop effective policies that are less burdensome to compute and implement.

Indeed, we believe that further characterization of the structure of optimal (type-H priority) routing policies is possible and would be of both theoretical and practical interest. We note that the reduced state space $S = \{0 \leqslant i + j \leqslant c\}$ used by the LP (4.23)–(4.30) corresponds to that of a loss system. Recent work on related loss systems such as, Altman et al. (1998), Savin et al. (2000), and Örmeci et al. (2000), suggests that δ-optimal routing policies for our system may be further characterized as state-dependent versions of a simple threshold policy. That is, given i type-H jobs are in service, there is a threshold number $j^*(i)$ such that it is optimal to put $(j^*(i) - j)^+$ type-L jobs immediately into service.

These state-dependent threshold policies require $O(c^2)$ decision variables, thereby reducing the computational complexity by an order of magnitude. The policies themselves are also far easier to implement. They require only $c+1$, rather than $c(c+1)/2$, parameters; for each i there is just one $j^*(i)$.

We also note that in systems in which at most one job may be routed at a time, this same order-of-magnitude reduction in the number of decision variables occurs. In these cases, for each of the $O(c^2)$ system states in which a job may be routed there are two decision variables. One represents the action of routing a type-L job (upon entry to the state) and the other the action of doing nothing.

Finally, length considerations for the paper have precluded us from including numerical investigations of the performance of the policies we have derived. It will be interesting to see how the throughput of type-L jobs changes with parameters such as the size of the system, c, and the relative load imposed by type-H calls, ρ, as well as by differences in the processing rates of the two types of work, μ_H and μ_L. It will also be interesting to understand how violations of the assumptions concerning stationarity and exponentiality affect the performance of the policies.

APPENDIX

The appendix can be found at the INFORMS home page in the Operations Research online collection at ⟨http://or.pubs.informs.org/Pages/collect.html⟩.

ACKNOWLEDGMENTS

The authors thank the referees and editors for their helpful comments. Research was supported by the Wharton Financial Institutions Center and by NSF grant SBR-9733739.

REFERENCES

Akşin, O. Z., P. T. Harker. 2001. Modeling a phone center: Analysis of a multi-channel, multi-resource processor-shared loss system. *Management Sci.* **47** 324–336.

Altman, E., A. Schwartz. 1991. Markov decision problems and state-action frequencies. *SIAM J. Control Optimiz.* **29** 786–809.

——, T. Jimenez, G. Koole. 1998. On optimal call admission control. *IEEE Trans. Commun.* **49** 1659–1668.

Armony, M., C. Maglaras. 2003. On customer contact centers with a call-back option: customer decisions, sequencing rules, and system design. *Oper. Res.*, to appear.

Avaya Communications. 2001. Blending: The changing color of contact center productivity. Retrieved September 9 ⟨http://www1.avaya.com/enterprise/whitepapers/gcc0910.pdf⟩.

Berman, O., R. C. Larson. 2000. A queuing control model for retail services having backroom operations and cross-trained workers. Working paper, Massachusetts Institute of Technology, Cambridge, MA.

Beutler, F. J., K. W. Ross. 1985. Optimal policies for controlled Markov chains with a constraint. *J. Math. Anal. Appl.* **112** 236–252.

Bhulai, S., G. Koole. 2000. A queueing model for call blending in call centers. *IEEE Trans. Auto. Control*, to appear.

Blanc, J., P. de Waal, P. Nain, D. Towsley. 1992. Optimal control of admission to a multiserver queue with two arrival streams. *IEEE Trans. Auto. Control* **37** 785–797.

Bolotin, V. A. 1994. Telephone circuit holding time distributions. *Proc. 14th International Teletraffic Conference*. Antibes Juan-les-Pins, France, 125–134.

Borst, S., A. Mandelbaum, M. I. Reiman. 2000. Dimensioning large call centers. Working paper, Technion, Haifa, Israel.

Brandt, A., M. Brandt. 1999. On a two-queue priority system with impatience and its application to a call center. *Methodology Comput. Appl. Probab.* **1** 191–210.

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao. 2002. Statistical analysis of a telephone call center: A queueing-science perspective. Working paper, The Wharton School, University of Pennsylvania, Philadelphia, PA.

Carr, S., I. Duenyas. 2000. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. *Oper. Res.* **48**(5) 709–720.

Durrett, R. 1996. *Probability: Theory and Examples*. Duxbury Press, Belmont, MA.

Fleischer, J., ed. 2000. One hundred cool things about call centers. *CallCenter Magazine* (October).

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

Garnett, O., A. Mandelbaum, M. I. Reiman. 2002. Designing a call-center with impatient customers. *Manufacturing Service Oper. Management* **4**(3) 208–227.

Genesys Telecommunications Laboratories. 2001. Genesys' universal queue^2 model for integrated media channels. Retrieved September 9 ⟨http://www.genesyslab.com/contact_center/solutions/universal_queue2.html⟩.

Green, L. V., P. J. Kolesar, J. Soares. 2001. Improving the SIPP approach for staffing service systems that have cyclic demands. *Oper. Res.* **49**(4) 549–564.

Guérin, R. 1998. Queueing-blocking system with two arrival streams and guard channels. *IEEE Trans. Comm.* **36** 153–163.

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

Jennings, O. B., A. Mandelbaum, W. A. Massey, W. Whitt. 1996. Server staffing to meet time-varying demand. *Management Sci.* **42**(10) 1383–1394.

Karlin, S., H. M. Taylor. 1975. *A First Course in Stochastic Processes*, 2nd ed. Academic Press, San Diego, CA.

Kleinrock, L. 1975. *Queueing Systems*, Volume 1. John Wiley and Sons, New York.

Mandelbaum, A. 2001. Private communication.

——, A. Sakov, S. Zeltyn. 2000. Empirical analysis of a call center. Working paper, Technion, Haifa, Israel.

Örmeci, E., A. Burnetas, J. van der Waal. 2000. Admission policies to a two-class loss system. *Comm. Statist.: Stochast. Models* **17**(4) 513–539.

Pinedo, M., S. Seshadri, J. G. Shanthikumar. 2000. Call centers in financial services: Strategy, technology, and operations. E. L. Melnick, P. R. Nayyar, M. L. Pinedo, S. Seshadri, eds. *Creating Value in Financial Services*. Kluwer Academic Publishers, Boston, MA.

Puhalskii, A. A., M. I. Reiman. 2000. The multiclass GI/PH/N queue in the Halfin-Whitt regime. *Advances Appl. Probab.* **32** 564–595.

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

Savin, S., M. Cohen, N. Gans, Z. Katalan. 2000. Capacity management in rental businesses with heterogeneous customer bases. Working paper, OPIM Department, The Wharton School, University of Pennsylvania, Philadelphia, PA.

Schaack, C., R. C. Larson. 1986. An N-server cutoff priority queue. *Oper. Res.* **34**(2) 257–266.

Sennott, L. I. 2001. Computing average optimal constrained policies in stochastic dynamic programming. *Probab. Engrg. Information Sci.* **15** 103–133.

Wolff, R. W. 1989. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, Englewood Cliffs, NJ.