Modeling and Optimization Problems in Contact Centers

Pierre L'Ecuyer
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal, H3C 3J7, Canada
Email: lecuyer@iro.umontreal.ca

Abstract

We give a quick overview of some key issues in (quantitative) call center management: building realistic models, developing efficient tools to simulate these models, finding quick approximation formulas for the performance measures of interest, and developing algorithms and software to optimize the staffing and scheduling of agents. This is discussed in the context of a multiskill center, in which different types of calls are handled by different agent groups (with different skill sets).

1. Introduction

Practically every large organization has a *contact center*, through which the customers or users can contact the organization and vice-versa, by telephone, FAX, email, Internet chat, and so on. When all contacts are made by telephone, we talk of a *call center*. The economic importance of call centers is greater than many would think: In North America, they employ around three percent of the workforce; this is more than in agriculture! The salaries of agents account for about 70% of the costs in typical call centers. Discussions of call center issues and models, and extensive reference lists, can be found in [7, 23, 38, 45, 48, 50], for example.

Based on forecasts of future call volumes, both in the short run and long run, call center managers must decide on the size and organization of their centers, plan the workforce (e.g., when to hire and train agents), decide on how many agents of each type to have in the center at each time to provide the required quality of service at minimal cost, construct working schedules for the available agents, select call routing strategies when there are different types of calls and agents, make outsourcing (subcontracting) decisions, and so on. All this decision-making gives plenty of opportunities for optimization.

The next section of this paper describes the main components (from a performance viewpoint) of a call center that handles multiple types of calls,. It points out some important modeling and optimization problems arising in this system and its variants. In Section 3, we discuss the development of realistic stochastic models for this system. The possibility of obtaining useful queueing formulas and approximations is briefly covered in Section 4. Tractable queueing models oversimplify reality and are not very reliable for many performance measures of interest in real-life call centers. In Section 5, we discuss simulation tools for realistic models of contact centers. Their efficiency is very important, because for realistic models, optimization must be done via simulation and this typically requires thousands of simulation runs. In Section 6, we mention important optimization problems encountered in this context, and discuss in more detail the staffing and scheduling of agents.

2. A Call Center with Multiple Call Types

2.1. Call types and agent skills

In a typical call center, the arriving calls are classified in different *types*, according to the required technical skill to answer the call, the language, importance of the call, etc. Agents are also classified in *skill groups* according to the subset of call types they can handle and (perhaps) their efficiency for handling these calls. Calls arrive at random according to some stochastic process. When a call arrives, it may be assigned immediately to an agent that can handle it (if there is one available) or it may be put in a queue (usually one queue per call type). When an agent becomes available, the agent may be assigned a call from one of the queues, or may remain idle (e.g., waiting for more important calls). All these assignments are made according to some *routing policy* that often incorporates *priority rules* for the calls and agents. Figure 1 illustrates this setting. We assume that there are K call types and I skill groups. In the figure, S_i represents skill group i, λ_k is the mean arrival rate for call type k, and $\mu_{k,i}$ is the mean service rate for call type k by an agent of group i. The *load* of call types k is

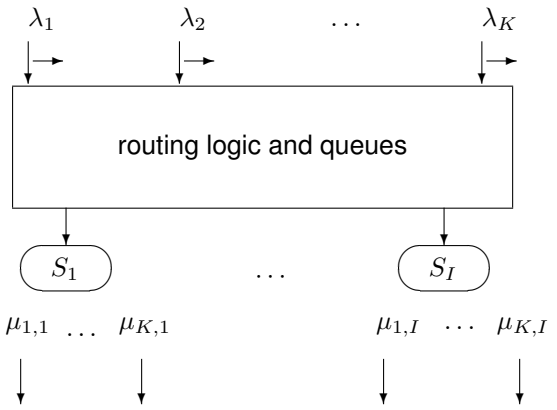

Figure 1. A multiskill call center

the total amount of agents' time required for their service; for example, if all are served by skill group i, their load is $\lambda_k/\mu_{k,i}$. Note that the arrival process is usually *not* a stationary Poisson process and the service times are usually *not* exponential.

Calls waiting in queue may *abandon* after a random *patience time* (this is represented by the horizontal arrows in the figure). Those who abandon may call again later, although those *retrials* are rarely modeled in practice, usually because of lack of sufficient data. Callers who received service may also call again for a number of reasons; these are called *returns*.

In the (degenerate) special case where each agent has a *single skill*, we have K single queues in parallel. If each agent has *all skills*, then we have a single skill set and a single queue. The system is obviously easier to analyze in these extreme cases. With all agents having all skills, the system is also more efficient (smaller waiting times, fewer abandonments) if we assume that the service time distribution for a given call type does not depend on the agent's skill set. However, this assumption turns out to be wrong in practice: agents are usually faster when they handle a smaller set of call types (even if their training gives them more skills). Agents with more skills are also more expensive; their salaries depend on their skill sets. Thus, for large-volume call types, it makes sense to dedicate a number of single-skill agents (specialists) to handle most of the load. A small number of agents with two or more skills can cover the fluctuations in the proportion of calls of each type in the arriving load. Wallace and Whitt [52] argue that for well-balanced systems, one or two skills per agent can often give a performance almost as good as all skills for all agents, even if we assume that the agents' speed does not depend on the skill set.

2.2. Performance measures

The main performance measures in call centers have to do with the quality of service and the operating costs. Here we only consider the cost of agents (their salaries). Most of the other costs are long-term investment costs and fixed costs (equipment, building rent, ...). In some cases, there are also communication costs (phone bills), outsourcing costs, penalty costs for bad service, and so on.

The quality of service in general is related to customer satisfaction. For example: Was the customer's concern or question resolved completely and quickly? Was the customer's experience pleasant? Here, we restrict ourselves to measures that depend on the waiting times of calls before they are answered.

The most frequently used measure in practice is the *service level* (SL), defined as the fraction of calls that wait less than an *acceptable waiting time* τ (typically 20 to 30 seconds). The SL can be measured and controlled separately by time period (hour, day, etc.) and by call type, or in an aggregated way. For example, one may ask that at least 80% of all calls must be answered within 20 seconds and that at least 90% of all calls of type i_0 (for a specific i_0) must be answered within 5 seconds. An important motivation for studying this measure is that for many types of call centers that provide services, in several countries, there are government regulations on the minimal acceptable SL and the call centers may have to pay very large fines when this SL is not met.

The SL is definitely not a perfect measure. In fact, to maximize the SL, an optimal policy would never serve any call that has already waited more than τ. But such an extreme policy is definitely unacceptable in most cases. A potential improvement would be to look at the *expected excess waiting time* over the threshold τ, defined as [37]:

$$\mathbb{E}[\max(0,\, W-\tau)] = \mathbb{E}[W-\tau \mid W>\tau]\cdot\mathbb{P}[W>\tau]$$

where W is the waiting time of a randomly chosen customer.

Another important measure is the *abandonment ratio*, defined as the fraction of calls that abandon; this could also be per call type, per period, and aggregated.

Managers also often look at the *occupation ratio* of agents, per agent type and per period. In theory, they would like to have their agents occupied as much as possible under the quality of service constraints. But important human factors must also be taken into account. Overstressed agents tend to perform more poorly in terms of both quality and speed. Generally, it is a bad idea to have occupation ratios above 90–92% for a sustained period of time. Another issue

is fairness between agents and agent groups: their occupation ratios should not differ too much.

In a center that brings revenue (e.g., via sales), other performance measures of interest could be the average revenue per agent per hour or the fraction of calls that lead to a sale.

2.3. Optimization and control

Management decisions are made at different time scales in contact centers. Long-term *strategic* decisions include choosing the size and layout of the center, the contact types to be handled, the skill groups, the types of work schedules, outsourcing decisions and contracts (if any), and so on. This requires long-term forecasts of call volumes. Hiring agents, and training new and old agents to increase their skills, is *medium-term tactical planning*. High turn-around of employees is frequent in call centers, so training is a (costly) continuous process. The *short-term planning* includes *staffing* decisions (how many agents of each type to have in the center at each time of the day), *scheduling* (how many agents should we have for each possible work schedule for the day or week and for each skill group), and *rostering* (assigning a working schedule to each physical agent, for each week). These decisions are made a few days to a few weeks in advance. Short-term *operational* decisions are made during the day. Hour-by-hour decisions include canceling training or meetings to have more agents available if the volume of calls is higher than expected, assigning agents to other tasks when the traffic is lower than expected, etc. Minute-by-minute decisions include the *routing rules* that assign that control in real time the agent-to-call and call-to-agent assignments. These rules act as controlling devices for the SLs, across call types and globally. They can be used for giving *priorities* to certain call types.

In a truly optimal routing policy, the decisions would generally depend on the entire state of the system. Such optimal *dynamic policies* are usually too complicated to compute and would be too hard to implement, although they can be computed or approximated in simple cases [10,25,39]. A class of commonly-used *static routing* rules operate as follows: Each call type has an ordered list of agent types; if all agents in the list are busy when a call of that type arrives, the call joins a queue (usually one queue per call type). Likewise, each agent type has an ordered list of queues (call types) to look for when it becomes available. This specifies some form of priorities [15, 22, 52]. One must be careful with such rules, because they could lead to imbalanced and sometimes unstable systems even if there is enough skill supply to cover the load [23]. If a call type has low priority in the ordered lists, it may happen that the agents that can serve it spend too much time on other call types and the fraction of their time left for the given call type does not suffice to handle the load. One may even construct examples where adding a skill to an agent *reduces* the overall performance of the system (because with the new skill, the agent may spend too much time on the wrong call type). To avoid these problems, we may prefer rules that look at the queue lengths or at the waiting time of the first call in queue for each queue, and use this information to select the queues.

In a *staffing problem*, the day is divided into periods (e.g, half hours) and the objective is to determine how many agents of each type to have in the center in each period, to meet the performance constraints (e.g., on the SL), at minimal cost. It is important to underline that the SL in one period can depend on the number of agents in other periods, either before or after. Suppose we can compute an optimal staffing solution. A major difficulty is that it is generally not possible to match exactly the optimal staffing by scheduling a set of agents whose working shifts are admissible (i.e., satisfy the constraints determined by Union agreements). The *scheduling problem* consists in determining a set of agents, each with its skill set and its (admissible) working schedule for the week (or working shift for the day), so that the performance constraints are met, at minimal cost. Of course, flexibility in the schedules may have a significant impact on the optimal cost. A *two-step approach* to this problem is to first compute an optimal staffing for each period and then to cover this staffing by a set of working shifts, at minimal cost. But this approach is generally suboptimal and the gap could be significant, e.g., if the skill sets of the agents selected in successive periods in the first step do not match. An optimal scheduling solution may also be impossible to implement, because in practice we do not have an infinite supply of each agent type. In a *scheduling and rostering problem*, for a given set of agents and a given set of admissible schedules, we must assign at most one schedule to each agent to meet the performance constraints at minimal cost.

2.4. Economies of scale and recourses to deal with uncertainty

Standard queueing formulas tell us that sharing resources provides economies of scale. For example, suppose two call centers handle the same types of calls and center j needs n_j agents to provide the required SL, for $j = 1, 2$. Then if we merge the two centers in a single one (and use good routing policies), the required number of agents cannot exceed $n_1 + n_2$, and is usually smaller. Larger call centers are generally more economical for this reason. Sharing the agents by merging the centers or by having agents with multiple skills can be seen as a recourse to deal with the uncertainty; it helps when there are more (or longer) calls than expected of a given type and less of another type in a given period of time.

For the situations where the *total volume* of calls in a

given time period can be much larger or much smaller than expected (i.e., has high variance), other types of recourses are needed. Flexibility to change the staffing levels on short notice can be very valuable if the daily call volume has high variability and is difficult to predict accurately, as is often the case [6]. For example, agents could be paid a certain amount to remain on standby at home, ready to be called for work at any time on a given day, and agents at work could be given the choice to go home when the traffic is too low. Another potential option is to sign an arrangement with an external provider to outsource the overflow on days of high volume. However, a major drawback of this type of option is the high cost of training these extra agents and the higher uncertainty in the quality of call handling by those agents whose services are required only a small fraction of the time.

Part of the agents' time in a call center is usually devoted to meetings and training sessions, which are normally scheduled every day. Meetings and training can be used as buffers against uncertainty: they can be canceled on days of high call volume and rescheduled (or recovered) on days of lower volume.

Blending is yet another way of buffering against uncertainty. In a blend system, inbound calls can be mixed with other types of contacts such as email, FAX, or outbound calls, which can be handled only when the traffic is lower. An *outbound call* is when the center tries to reach a customer. This could be because this customer has left a message and wants to be called, or because of specific problems with this particular customer (unpaid bill, etc.), but most often the outbound calls are marketing tools: the center wants to reach customers to sell them products. In a blend center with inbound and outbound calls, the outbound calls are triggered by a *predictive dialer*; this is an automatic system that dials customer numbers (usually several numbers in parallel) when the traffic is deemed low enough, trying to reach customers. A *right party connect* occurs when the outbound contact is successful. A *mismatch* occurs when the successful contact cannot be served immediately because no agent is available (this can happen if a larger than expected fraction of the dials were successful or if a large number of inbound calls arrived in the meantime). In this context, we may want to put a lower bound on the expected volume of outbound calls per day, or on the expected volume of sales per day from the outbound calls (the probability of reaching a customer and the probability of a sale may depend on the time of day), and an upper bound on the expected number (or fraction) of mismatches. We may then want to optimize the operating cost of the center under these additional constraints. This gives rise to challenging optimization problems.

3. Realistic Modeling

Building realistic models of contact centers is difficult mainly because of the lack of detailed information and data. In most cases, the data is aggregated automatically by the computerized systems that collect it. It is not uncommon to have only averages over each half-hour of the day; e.g., the total number of arrivals, the number of abandonments, the average service time, the number of calls with good SL, and perhaps a few other measures, for each call type and each half-hour. For the agent groups, we may have their number and occupancy ratio over each half-hour. It is difficult to find the appropriate distributions and dependencies between random variables with such aggregated data. There is typically very little information available (if any) on the patience-time distribution and on the retrials, for example. Agents might not be logged-in to the system 100% of the time for a number of reasons, and there is often little or no information to model this. Call center managers often have their own (undocumented) ad-hoc rules to control the system when there is a change in the total call volume, or call type proportions, or if the SL goes down too much. Most models neglect these important aspects. Nevertheless, enough information has been found in the available data to support the following conclusions.

The *arrival process* of calls is *not* a Poisson process with deterministic (time-dependent) rate. In all studies that we know, the arrival process agrees with a Poisson process only if the arrival rate of the Poisson process is itself a stochastic process [6, 13, 36, 51]. Typically, the variance of the total number of calls in a given time period is much larger than the mean (for a Poisson process, it should be equal to the mean). The (mean) arrival rate also depends strongly on the time-of-day and often on the day-of-week. Finally, there is positive stochastic dependence between arrival rates in successive periods within a day, and between arrival volumes of successive days. Models that agree with these properties are proposed by [6].

Service time distributions are often assumed to be exponential and some studies have found this reasonable [30]. However, recent call center data analyzes have found that, among the parametric distributions, the lognormal is usually a much better fit [13, 19, 21, 50]. Christos Alexopoulos (personal communication) recently found a case where the lognormal was not appropriate but the log-logistic was an excellent fit.

The *patience-time distribution* is important to model correctly, because it can have a significant impact on the SL and abandonment ratio, especially in heavy-traffic conditions [23, 56]. The data on time to abandonment (when available) is highly censored, because the large majority of call (usually more than 95%) are answered before abandonment. Thus, estimating the patience-time distribution re-

quires the use of special statistical techniques for censored data [13].

There are very few studies on retrial estimation. One of them is [33], where the first-call arrival rates and retrial rates are estimated jointly from detailed data. Ignoring retrials can lead to significant modeling errors, in particular because in heavy-traffic conditions, the retrials amplify the observed (net) arrival rate [1].

4. Queueing Approximations

In the case of a single call type, if we assume that the system is in steady-state, the arrival process is Poisson with constant rate λ, the service times are independent and exponential with rate μ, there are no abandonments, and there are s agents, then we have a standard M/M/s queue, for which the *delay probability* $\mathbb{P}[W > 0]$ (where W is the waiting time of a randomly selected customer) is given by the so-called Erlang C formula [20, 23]. Given that a customer must wait, its conditional waiting time in queue is exponential with known mean $1/(s\mu - \lambda)$. The SL for a given s is easy to compute from this information:

$$\mathbb{P}[W > \tau] = \mathbb{P}[W > 0] \exp[(s\mu - \lambda)\tau].$$

The minimal s required to reach a given SL can be obtained via some root finding method, using the fact that the SL is monotone in s.

Halfin and Whitt [29] have developed an approximation to the Erlang C formula when s is large and the system has high utilization, whereas $\mathbb{P}[W > 0]$ has a predetermined value. It is much simpler to compute and provides better insight than the exact formula. They consider a sequence of M/M/s queues for which $s \to \infty$ and $(1 - \rho)\sqrt{s} \to \beta$ for $0 < \beta < 1$, where μ is fixed and $\rho = \lambda/(s\mu)$ is the utilization factor of the system, and prove that

$$\mathbb{P}[W > 0] \to \frac{1}{1 + \beta\Phi(\beta)/\phi(\beta)} \qquad (1)$$

where Φ and ϕ are the standard normal distribution function and density, respectively. This approximation has been extended in [55] and is further discussed in [12, 23]. The *square-root safety staffing* (approximation) formula commonly used in call centers is justified by this result. This formula works as follows: To achieve a given target α for the delay probability $\mathbb{P}[W > 0]$, for a load $r = \lambda/\mu$, we take $s = r + \Delta$ agents, where $\Delta = \beta\sqrt{r}$ is the "safety staffing" above the load to account for the stochastic variability. To obtain this formula, it suffices to multiply the approximation $(1 - r/s)\sqrt{s} \approx \beta$ by $\sqrt{s}$ and note that $\sqrt{s}/\sqrt{r} \to 1$ as $s \to \infty$. The formula gives insight in the economy of scale made when the load increases: the safety staffing is proportional to the square root of the load. When expressed as a percentage of the load, this percentage goes to zero as $1/\sqrt{r}$ when $r \to \infty$.

In the Halfin-Whitt asymptotic regime, s and r go to infinity simultaneously at the same rate and the delay probability converges to a constant. Other types of heavy-traffic limits have been defined in the literature. In the so-called *conventional* regime, for example, s is kept constant whereas both λ and μ increase linearly so that ρ and $\mathbb{P}[W > 0]$ both converge to 1 [54]. If ρ is fixed while λ and s increase to infinity, then $\mathbb{P}[W > 0]$ converges to 0, so in the limit, nobody waits. This *quality-driven* asymptotic regime is appropriate for call centers where speed of answer is much more important than the cost of agents (think of emergency services, for instance). If we fix the safety staffing Δ while both λ and s increase to infinity, then $\mathbb{P}[W > 0]$ converges to 1, i.e., everyone waits in the limit. This is an *efficiency-driven* regime. It is appropriate for situations where the productivity of agents is much more important than the wait (think of answering emails or processing orders made by Internet, for example). An asymptotic regime for which $\mathbb{P}[W > 0]$ is fixed to a constant in the interval (0,1), such as the Halfin-Whitt regime, is called a *quality and efficiency-driven* (QED) regime. This type of regime is in good agreement with the dynamics of a typical large call center with constraints on the SL and abandonment ratio. These regimes are further discussed in [54].

The formulas and approximations just described have counterparts for the case where there are abandonments, under the assumption that the patience times are exponential. The Erlang C model is then replaced by the Erlang A model (we obtain an M/M/s+M queue, i.e., with Markovian abandonment) [14,23,26,47]. There is still a square-root formula for the safety staffing in the QED regime, but the constant β that corresponds to a given value of the delay probability is *smaller* than that given by (1) and could even be negative if there is a large percentage of abandonments [26]. Moreover, the percentage of abandonments, the mean service time, and the server's idleness are all in $O(1/\sqrt{s})$ in the QED regime. In the extreme case where the abandonment rate goes to infinity, we have a system where every waiting customer abandons. This type of *loss system* corresponds to the Erlang B model.

To accommodate *time-dependent arrival rates*, the standard procedure is to partition the time in short periods (e.g., half-hours), assume that the system is stationary with a given arrival rate over each period, and apply the Erlang models separately for each period. This neglects transient effects. This type of piecewise stationary approximation can be taken to the limit with the length of the periods going to zero; the arrival rate can vary continuously in time, the Erlang formulas are used at each point in time, and the results are integrated with respect to time [28, 53]. This is *pointwise stationary approximation*. An improvement is to

use some *delay* Δ to account for the transient effect: the arrival rate at time t is used to compute the performance at time $t + \Delta$ [27, 34]. Other methods are discussed and compared in [34].

All these approximations are for a single call type only. Queueing approximations for the SL in multiskill systems are much more difficult to obtain.

In the context of a call center with multiple call types, multiple skills, and doubly stochastic time-varying arrival processes, Bassamboo et al. [9] study an asymptotic regime where the arrival rates increase faster than linearly, the service and abandonment rates increase linearly, and the number of servers also increases toward infinity. They show how to compute a staffing and routing policy that minimizes a the costs of staffing and abandonments (assumed to be linear in the service capacity and abandonment counts) in their limiting regime.

In the case of *loss systems* (no waiting queues, calls that cannot be served immediately are lost), good approximations of the loss (or *blocking*) probability per call type have been obtained via a two-moment approximation of overflow process (the *equivalent random method*, or Hayward's approximation) [17,18,20,57] and using an hyper-exponential approximation [22]. Loss systems are much easier to analyze than systems with queues, mainly because they have a much smaller state space, and a strong correlation can be observed between average waiting time or the loss probability in certain queueing systems and the loss probability in the corresponding loss system.

Attempts to extend these types of approximations for estimating the SL in systems where there are waiting queues and fixed routing rules can be found in [5,40,41]. Although these approximations of the SL are often far from realistic, they turn out to be useful as rough-cut approximations to find a good starting solution in the first stage of a staffing optimization algorithm [5]. But to obtain reliable estimates of the SL, abandonment ratios, server utilization, etc., in a multiskill center, one must use simulation.

5. Simulation Tools

In view of the large gap between call-center reality and approximate queueing models that are tractable, simulation is a key tool for accurate performance estimation and for optimization [45, 48]. Once the model is well-defined (see Section 3), there is no fundamental difficulty in principle to incorporate its detailed stochastic behavior in a simulation program. But the programming effort can be substantial if the model is complicated and the execution time can also be excessive, e.g., if the call center is large, each call is simulated in detail, and the routing policy is complicated. Fast simulation is especially important if the simulation is to be used inside an optimization algorithm (so thousands of simulations at different parameter settings could be needed).

There is commercial software for detailed simulation of contact centers; for example Rockwell's *Arena Contact Center Edition* [8] and NovaSim's *ccProphet* [49]. These products offer the advantages of a nice graphical interface, graphical animation, and permit one to build and run a simulation program without explicit programming. On the other hand, they are expensive to buy, the execution is rather slow, and the tools are often not flexible enough to easily model the complexities of call centers (e.g., the routing policies in multiskill centers) and to integrate variance reduction methods to make the simulation statistically more efficient.

To overcome these problems, Buist and L'Ecuyer have developed a Java library for the simulation of contact centers, called the *ContactCenters* library [15]. see also `http://www.ericbuist.com/contactcenters`. It is based on the well-supported modern programming language Java and is built over the SSJ simulation library [42, 43]. Since it not depend on a specific graphical interface, we were able to make it much more flexible than commercial products.

Interoperability with other software (statistics, optimization, databases, etc.) is easily achieved via the Java interfaces with this software. The library provides building blocks to simulate all kinds of call centers. It supports several types of contacts, multiskill, blend, arbitrary dialing and routing policies, various types of arrival processes, and so on. Execution times are faster than with commercial software. We tried the examples provided with Arena Contact Center Edition and they ran about 30 times faster with our Java library than with Arena.

The flexibility offered by the library facilitates the implementation of variance reduction techniques and gradient (or subgradient) estimators [3, 16]. Specific variance reduction techniques for the simulation of a multiskill call center with stochastic arrival rate are studied and experimented in [44].

A (quite general) generic model of a multiskill contact center, preprogrammed in Java, comes with the library. The model parameters can be specified in an XML file and simulations can be run either from another program (e.g., when doing optimization) or just by running an executable (no need to program).

6. Optimization

Here we concentrate on the staffing and scheduling. Optimization of routing rules was discussed briefly in Section 2.3. At a slightly higher level, the optimization of *outsourcing*, *hiring*, and *training* decisions can be formulated as a stochastic control problem which could be solved (at least in principle) by dynamic programming [24]. Good

forecasts of the arrival rates (workload prediction) are required before solving any of these problems.

The following scheduling optimization problem for one day of operation of a multiskill center is taken from [16]. Single-skill versions of this problem were studied earlier in [4, 35]. There are K call types, I skill groups, P time periods, and Q types of working *shifts* for the day. In general, a *schedule* specifies the working hours of an agent over several days (e.g., a week), and a *shift* specifies them over a single continuous presence at the working place (e.g., a day), but here we consider a single day, so a shift and a schedule are the same. The *cost vector* is $\mathbf{c} = (c_{1,1}, \ldots, c_{1,Q}, \ldots, c_{I,1}, \ldots, c_{I,Q})^{\mathrm{t}}$, where "t" means "transposed" and $c_{i,q}$ is the cost of an agent of type i having shift q. The *decision variables* are $\mathbf{x} = (x_{1,1}, \ldots, x_{1,Q}, \ldots, x_{I,1}, \ldots, x_{I,Q})^{\mathrm{t}}$, where is $x_{i,q}$ the number of agents of type i having shift q. The *staffing vector* is $\mathbf{y} = (y_{1,1}, \ldots, y_{1,P}, \ldots, y_{I,1}, \ldots, y_{I,P})^{\mathrm{t}}$ where $y_{i,p}$ is the number of agents of type i in period p. This vector $\mathbf{y}$ can be written as $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a block diagonal matrix with I blocks $\tilde{\mathbf{A}}$, where the element (p, q) of $\tilde{\mathbf{A}}$ is 1 if shift q covers period p, and 0 otherwise. The abandonments and waiting times of the calls depend on $\mathbf{x}$ only via $\mathbf{y}$. The SL for call type k and period p is

$$g_{k,p}(\mathbf{y}) = \frac{\mathbb{E}[\text{Num. calls ans. within } \tau_{k,p} \text{ sec in period } p]}{\mathbb{E}[\text{Num. calls arriving in period } p]}$$

for some constant $\tau_{k,p}$, where the expectations are with respect to the probability distributions that define the stochastic model (arrival rate process, service times, abandonments, etc.). Similarly, the aggregate SL over call type k is the expected total number of calls of type k answered within some time limit τ_k over the day (say), divided by the expected total number of calls of type k received over the day. We denote by $g_p(\mathbf{y})$, $g_k(\mathbf{y})$ and $g(\mathbf{y})$ the aggregate SLs for period p, call type k, and overall, respectively. The corresponding time limits are τ_p, τ_k, and τ, and the corresponding minimal SLs are l_p, l_k and l.

The *scheduling optimization problem* with SL constraints can then be formulated as

$$\begin{array}{rll} \min & \mathbf{c}^{\mathrm{t}}\mathbf{x} = \sum_{i=1}^{I}\sum_{q=1}^{Q} c_{i,q}x_{i,q} & \\ \text{subject to} & g_{k,p}(\mathbf{A}\mathbf{x}) \geq l_{k,p} & \text{for all } k, p, \\ & g_p(\mathbf{A}\mathbf{x}) \geq l_p & \text{for all } p, \\ & g_k(\mathbf{A}\mathbf{x}) \geq l_k & \text{for all } k, \\ & g(\mathbf{A}\mathbf{x}) \geq l, & \\ & \mathbf{x} \geq 0, \text{ and integer.} & \end{array} \tag{P1}$$

Assume now (cavalierly) that any staffing $\mathbf{y}$ is admissible and that an agent of group i in period p costs $c_{i,p}$. Denoting $\mathbf{c} = (c_{1,1}, \ldots, c_{1,P}, \ldots, c_{I,1}, \ldots, c_{I,P})^{\mathrm{t}}$, this gives the following *staffing problem*, which is a *relaxation* of (P1):

$$\begin{array}{rll} \min & \mathbf{c}^{\mathrm{t}}\mathbf{y} = \sum_{i=1}^{I}\sum_{p=1}^{P} c_{i,p}y_{i,p} & \\ \text{subject to} & g_{k,p}(\mathbf{y}) \geq l_{k,p} & \text{for all } k, p, \\ & g_p(\mathbf{y}) \geq l_p & \text{for all } p, \\ & g_k(\mathbf{y}) \geq l_k & \text{for all } k, \\ & g(\mathbf{y}) \geq l, & \\ & \mathbf{y} \geq 0, \text{ and integer.} & \end{array} \tag{P2}$$

A further simplification of (P2) arises by considering a *single period* ($P = 1$).

To solve any of these problems, we need to approximate or estimate the SL functions $g_\bullet$. In general, *each* of these functions may depend on *several* components of the vector $\mathbf{y}$ in a complicated way, because changing the number or mix of agents in one period may change the queue length in the periods that follow, for example, or may change the waiting time of a call that arrived in the previous period. Simplified queueing models may sometimes provide reasonable rough-cut approximations to these SL functions, but simulation seems to be the only way of getting reliable estimates of their values for realistic call centers.

Simulation-based methods combined with integer programming have been used to solve instances of (P2) with either a single skill [4] or a single period [16]. These authors start with a relaxation of the integer program and add cuts (linear constraints), derived from the SL constraints that are not satisfied, to drive the solution toward feasibility. For large problems, they solve the linear program instead of the integer program and round up the solution (the variables $x_{i,q}$ or $y_{i,p}$) to the next integer. A realistic problem instance with 65 call types and 89 skill sets is solved in [16]. However, the delivered solution is often suboptimal by a few percentage points, mainly because of the noise in the simulation, and takes several minutes to compute. Different solution methods are proposed in [35], where the SL is approximated by transient analysis of a continuous-time Markov chain instead of by simulation, and in [5], where the SL is approximated by a heuristic called the *loss-delay* approximation and a near-optimal solution is found using neighborhood search. In [5], simulation is used at the end to correct the approximation and refine the solution. The resulting method is competitive with that of [16]; it often performs better when operating under a limited computing budget. A fast two-step algorithm for shift scheduling in multiskill centers, with aggregated SL constraints across all call types only, is proposed in [11]. In the first step, the method computes an optimal staffing for each time period. In the second step, it solves a linear program to find a set of shifts that cover this staffing, by allowing agents to use

only a subset of their skills in certain periods if needed. This type of two-step approach may lead to suboptimal solutions of the original scheduling problem.

A key ingredient for the efficiency of the methods based on relaxed integer programming is a good set of initial constraints to define a small region that contains the feasible set. For the methods based on neighborhood search, it is important to start from a good initial solution. Such sets of constraints or initial solutions can be obtained by using the following observations (and their refinements).

Consider a single time period. If the agents from skill group i handle a fraction $\beta_{i,k}$ of the type-k calls, then $r_{i,k} = \beta_{i,k}\lambda_k/\mu_{i,k}$ is the *load* of call type k that goes to skill group i. If there are x_i agents in group i and if there are no abandonments, then we must obviously have $x_i \geq \sum_{k=1}^{K} r_{i,k}$ for all i, otherwise the system is clearly unstable. In other words, we must have enough *skill supply* to cover the load, for each i. Based on this, a naive way to do the staffing would be to solve a linear optimization problem that minimizes the agents' costs under the above inequality constraints, with real variables $\beta_{i,k} \geq 0$ and integer variables $x_{i,k} \geq 0$. However, barely covering the load is *not sufficient* to meet the SL constraints, because of the stochasticity of the system. If there are abandonments, then it may also be unnecessary to cover all the load.

To get a more accurate picture, in the formula given above for the required skill supply, a *safety staffing* should be added to the total load $\sum_{k=1}^{K} r_{i,k}$ that must be covered. This could be computed by inverting the Erlang-C or Erlang-A formula, or by using an appropriate QED-regime approximation, for each pair (i, k). This would provide a set of constraints that can reduce in a useful way the set of feasible solutions to be considered in an optimization problem.

However, the $r_{i,k}$'s depend in turn jointly on the routing strategy and staffing decisions. Ideally, one would like to optimize the routing and staffing/scheduling simultaneously. This gives rise to large and complex optimization problems for which only special cases have been studied so far; see, e.g., [2, 9, 31, 32, 46].

Acknowledgments

This work has been supported by a grant from Bell Canada via the Bell University Laboratories, Grants No. CRDPJ-251320 and ODGP0110050 from NSERC-Canada, and a Canada Research Chair to the author. The authors thanks Christos Alexopoulos, Eric Buist, Vijay Mehrotra, Ornella Pisacane, and Auke Pot for their helpful suggestions.

References

[1] M. S. Aguir, O. Akşin, F. Karaesmen, and Y. Dallery. The impact of retrials on call center performance. *OR Spectrum*, 26(3):353–376, 2004.

[2] R. Atar, A. Mandelbaum, and M. I. Reiman. Scheduling a multi class queue with many exponential servers: Asymptotic optimality in heavy traffic. *Annals of Applied Probability*, 14:1084–1134, 2004.

[3] J. Atlason, M. A. Epelman, and S. G. Henderson. Using simulation to approximate subgradients of convex performance measures in service systems. In S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, editors, *Proceedings of the 2003 Winter Simulation Conference*, pages 1824–1832. IEEE Press, 2003.

[4] J. Atlason, M. A. Epelman, and S. G. Henderson. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research*, 127:333–358, 2004.

[5] A. N. Avramidis, W. Chan, and P. L'Ecuyer. Staffing multi-skill call centers via search methods and a performance approximation. Manuscript, 2006.

[6] A. N. Avramidis, A. Deslauriers, and P. L'Ecuyer. Modeling daily arrivals to a telephone call center. *Management Science*, 50(7):896–908, 2004.

[7] A. N. Avramidis and P. L'Ecuyer. Modeling and simulation of call centers. In *Proceedings of the 2005 Winter Simulation Conference*, pages 144–152. IEEE Press, 2005.

[8] V. Bapat. The Arena product family: Enterprise modeling solutions. In S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, editors, *Proceedings of the 2003 Winter Simulation Conference*, pages 210–217. IEEE Press, 2003.

[9] A. Bassamboo, J. M. Harrison, and A. Zeevi. Design and control of a large call center: Asymptotic analysis of an LP-based method. Manuscript, Graduate School of Business, Stanford University, 2004.

[10] S. Bhulai and G. Koole. A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control*, pages 1434–1438, 2003.

[11] S. Bhulai, G. Koole, and G. Pot. Simple methods for shift scheduling in multi-skill call centers. Technical report, Technical Report WS 2005-10, Free University, Amsterdam, 2005.

[12] S. Borst, A. Mandelbaum, and M. Reiman. Dimensioning large call centers. *Operations Research*, 52:17–34, 2004.

[13] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100:36–50, 2005.

[14] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100:36–50, 2005.

[15] E. Buist and P. L'Ecuyer. A Java library for simulating contact centers. In *Proceedings of the 2005 Winter Simulation Conference*, pages 556–565. IEEE Press, 2005.

[16] M. T. Ceẓik and P. L'Ecuyer. Staffing multiskill call centers via linear programming and simulation. *Management Science*, 2006. To appear.

[17] P. Chevalier, R. A. Shumsky, and N. Tabordon. Overflow analysis and cross-trained servers. *International Journal of Production Economics*, 85:47–60, 2003.

[18] P. Chevalier, R. A. Shumsky, and N. Tabordon. Routing and staffing in large call centers with specialized and fully flexible servers. Technical report, Simon Graduate School of Business, University of Rochester, 2004.

[19] E. Chlebus. Empirical validation of call holding time distribution in cellular communications systems. In *Proceedings of the 15th International Teletraffic Congress*, pages 1179–1188. Elsevier, 1997.

[20] R. B. Cooper. *Introduction to Queueing Theory*. North-Holland, New York, second edition, 1981.

[21] A. Deslauriers. Modélisation et simulation d'un centre d'appels téléphoniques dans un environnement mixte. Master's thesis, Department of Computer Science and Operations Research, University of Montreal, Montreal, Canada, 2003.

[22] G. J. Franx, G. Koole, and A. Pot. Approximating multi-skill blocking systems by hyper-exponential decomposition. *Performance Evaluation*, 63:799–824, 2006.

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

[24] N. Gans and Y.-P. Zhou. Managing learning and turnover in employee staffing. *Operations Research*, 50:991–1006, 2002.

[25] N. Gans and Y.-P. Zhou. Overflow routing for call center outsourcing, 2004. Preprint.

[26] O. Garnett, A. Mandelbaum, and M. Reiman. Designing a call center with impatient customers. *Manufacturing and Service Operations Management*, 4(3):208–227, 2002.

[27] L. V. Green and P. Kolesar. The lagged PSA for estimating peak congestion in multiserver Markovian queues with periodic arrival rates. *Management Science*, 43:80–87, 1997.

[28] L. V. Green and P. J. Kolesar. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science*, 37(1):84–97, 1991.

[29] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–588, 1981.

[30] C. Harris, K. Hoffman, and P. Saunders. Modeling the irs telephone taxpayer information system. *Operations Research*, 35:504–523, 1987.

[31] J. M. Harrison and M. J. Lopez. Heavy-traffic resource pooling in parallel-server systems. *Queueing Systems*, 33:339–368, 1999.

[32] J. M. Harrison and A. Zeevi. Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy-traffic regime. *Operations Research*, 52:243–257, 2004.

[33] K. L. Hoffman and C. M. Harris. Estimation of a caller retrial rate for a telephone information system. *European Journal of Operational Research*, 27(2):207–214, 1986.

[34] A. Ingolfsson, E. Akhmetshina, S. Budge, Y. Li, and X. Wu. A survey and experimental comparison of service level approximation methods for non-stationary $m/m/s$ queueing systems. Technical report, School of Business, University of Alberta, Edmonton, Alberta, Canada, 2005.

[35] A. Ingolfsson, E. Cabral, and X. Wu. Combining integer programming and the randomization method to schedule employees. Technical report, School of Business, University of Alberta, Edmonton, Alberta, Canada, 2003. Preprint.

[36] G. Jongbloed and G. Koole. Managing uncertainty in call centers using Poisson mixtures. *Applied Stochastic Models in Business and Industry*, 17:307–318, 2001.

[37] G. Koole. Redefining the service level in call centers. Technical report, Department of Stochastics, Vrije Universiteit, Amsterdam, 2003.

[38] G. Koole and A. Mandelbaum. Queueing models of call centers: An introduction. *Annals of Operations Research*, 113:41–59, 2002.

[39] G. Koole and A. Pot. Approximate dynamic programming in multi-skill call centers. In *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, 2005.

[40] G. Koole, A. Pot, and J. Talim. Routing heuristics for multi-skill call centers. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1813–1816. IEEE Press, 2003.

[41] G. Koole and J. Talim. Exponential approximation of multi-skill call centers architecture. In *Proceedings of QNETs*, pages 23/1–10, 2000.

[42] P. L'Ecuyer. *SSJ: A Java Library for Stochastic Simulation*, 2004. Software user's guide, Available at `http://www.iro.umontreal.ca/~lecuyer`.

[43] P. L'Ecuyer and E. Buist. Simulation in Java with SSJ. In *Proceedings of the 2005 Winter Simulation Conference*, pages 611–620. IEEE Press, 2005.

[44] P. L'Ecuyer and E. Buist. Variance reduction in the simulation of call centers. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE Press, 2006. To appear.

[45] A. Mandelbaum. Call centers: Research bibliography with abstracts, 2006. Version 7, downloadable from `http://iew3.technion.ac.il/serveng/References/references.html`.

[46] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized c-mu rule. *Operations Research*, 52:836–855, 2004.

[47] A. Mandelbaum and S. Zeltyn. Service engineering in action: The Palm/Erlang-A queue, with applications to call centers. Manuscript, downloadable from `http://iew3.technion.ac.il/serveng/References/references.html`.

[48] V. Mehrotra and J. Fama. Call center simulation modeling: Methods, challenges, and opportunities. In *Proceedings of the 2003 Winter Simulation Conference*, pages 135–143. IEEE Press, 2003.

[49] NovaSim. ccProphet — simulate your call center's performance, 2003. See `http://www.novasim.com/CCProphet/`.

[50] J. Pichitlamken, A. Deslauriers, P. L'Ecuyer, and A. N. Avramidis. Modeling and simulation of a telephone call center. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1805–1812. IEEE Press, 2003.

[51] S. G. Steckley, S. G. Henderson, and V. Mehrotra. Service system planning in the presence of a random arrival rate, 2004. submitted.

[52] R. B. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. working paper, available at http://www.columbia.edu/~ww2040/poolingMSOMrevR.pdf, 2005.
[53] W. Whitt. The pointwise stationary approximation for $M(t)/M(t)/s$ queues is asymptotically correct as the rates increase. *Management Science*, 37(3):307–314, 1991.
[54] W. Whitt. *Stochastic-Process Limits*. Springer-Verlag, New York, 2001.
[55] W. Whitt. A diffusion approximation for the G/GI/n/m queue. *Operations Research*, 6:922–941, 2004.
[56] W. Whitt. Fluid models for many-server queues with abandonments. *Operations Research*, 2004. To appear.
[57] R. W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice-Hall, New York, 1989.