Technical Report UCSC-CRL-07-04: Call Center Routing Strategies in the Presence of Servers with Heterogeneous Performance Attributes

Vijay Mehrotra
San Francisco State University
San Francisco, CA

Kevin Ross
University of California
Santa Cruz, CA

Yong-Pin Zhou
University of Washington
Seattle, WA 98195-3200

September 27, 2007

Abstract

First call resolution, which in essence means the proportion of inquires that are successfully addressed after one call (note that the definitions of FCR differ, see below), has been getting more attention in call center management. A review of the literature, however, reveals that most of the interest has come from the practitioners (call center managers, consultants, etc.). We can only find a few research reports on FCR - the benefits, the potential downsides, and more importantly, how FCR should be implemented in the routing of calls.

1 Introduction

Over the past two decades, customer service call centers have become a very important part of many companies' business operations. Today, inbound call centers employ millions of agents across the globe and serve as a primary customer-facing channel in many different industries. As such, there has been a great deal of research interest in call center operations management (Gans, Koole and Mandelbaum [9] and Aksin, Armony, and Mehrotra [1] provide very thorough literature surveys).

Much of this research has focused on queueing models, staffing, and performance analysis. For example, one common operational setting is a call center in which there is a single type of inbound

call (which we refer to as the Single Queue model). In this setting, a key operational challenge is the determination of how many agents to staff in order to achieve a target mean waiting time (which is often referred to as Average Speed of Answer or ASA) or waiting time distribution (which is typically represented by an objective of at least some fixed percentage of calls within some target time period and referred to as Service Level.

For models in which there are multiple types of inbound calls, on the other hand, the performance analysis and staffing problems become significantly more challenging when some or all of the agents are able to handle more than one type of call. This latter setting is often referred to as Skill-Based Routing, because calls are routed to different agents (or groups of agents) based on logic that takes into account which agents are capable of handling which types of calls. The challenge in this setting is to simultaneously determine how many agents should be staffed and which skills and priorities each agent should be assigned in order to achieve particular ASA or Service Level targets for each queue.

Historically, the vast majority of the research literature has used either ASA and Service Level as the primary performance metric with which to judge a particular staffing configuration for both the Single Queue and the Skill-Based Routing settings. This is because customer waiting time has historically been viewed as a proxy for a customer's satisfaction with the service delivered by the call center, since it is widely agreed that customers prefer to spent little or no time waiting for service.

More recently, some researchers have begun to model customer reneging, which in the call center context is typically referred to as Abandonment, and to include the customer Abandonment Rate as an important metric in evaluating operational performance (see Mandelbaum and Zeltyn [15] for a good survey of the state of the art in this area). There are two main reasons for including customer abandonment in call center models. First of all, customers who abandon the queue are quite likely dissatisfied with the service encounter, and therefore this metric is an important one for call center managers who are focused on delivering high-quality customer service. Secondly, the effect of customer abandonment is to reduce the total traffic in the call center, and thus abandonment can have a significant impact on staffing needs and on customer waiting times.

It is important to note that ASA, Service Level, and Abandonment Rates are all metrics that are based on a customer's waiting experience prior to service. However, it is well known in the marketing and customer satisfaction literature that the customer's experience during service is also a very strong determinant of customer satisfaction and loyalty. In particular, a second call from

a customer about a specific issue is a clear sign that the issue had not been resolved during the previous service encounter, and this lack of resolution is a strong sign of customer dissatisfaction. Thus, the Call Resolution (CR) rate and the First Call Resolution (FCR) rate[1] are very important customer-centric metrics that have been largely ignored in the operations management literature.

In many call centers, agents have been trained to handle all calls within a particular queue but nevertheless exhibit very different performance across specific types of calls within that queue, where performance is defined by average call handling time (AHT) and first call resolution rate (FCR). A good overview of routing in multi-skill environments is found in [14], with efficient policies described in [3]. Worker cross-training is known to improve those performance measures [16], but here we assume rates to be fixed for the period of routing decisions.

In this paper, we explore strategies for determining which calls within a queue should be handled by which agents, where these assignments are made dynamically based on the specific attributes of the agents and/or the current state of the system. In particular, each call type may be handled by different agent types with different AHT and CR rate; and each agent type may handle different call types with different AHT and CR. In practice one can find cases where AHT and CR are positively correlated (the faster one works, the sloppier the job) or negatively correlated (more experienced and better trained agents can handle calls both faster and better). For more discussions on this please refer to de Véricourt and Zhou [7]. In our paper we do not assume any specific form of correlation between the two; any relation is possible.

We believe that this paper makes several important contributions to the call center operations management literature. First of all, we present a richer framework for call routing than the traditional FIFO call center model while using actual performance data in two very important dimensions (AHT and CR) as the basis for call assignment decisions. Secondly, we formulate a mathematical program that provides a quick analytic measure of the maximum First Call Resolution rate under very reasonable constraints on agent utilization. Thirdly, we develop a variety of intelligent routing rules that are intended to deliver both low ASA values and high CR rates, where several of these routing rules are based on outputs from our optimization model. Finally, we conduct empirical tests of these rules against operational data from a large financial service firm's customer service call centers.

[1]Our model uses the CR measure since we don't track the number of time the customer has sought help with the same problem before, for analytical tractability. FCR is the term most practitioners use, however. In this paper we will use CR when describing our model, and FCR when referring to its usage in practice and in the literature.

The remainder of this paper is organized as follows. In Section 2, we present a survey of the research literature on models that take into account call resolution rates and customer callbacks. In Section 3, we then formulate a mathematical program that can be solved to determine the optimal long-term proportion of calls of each type that should be handled by each group of agents in order to maximize the overall CR rate. In Section 4, we develop several routing strategies designed to deliver a higher CR rate than the traditional FIFO model while also seeking to keep customer waiting times as low as possible, where some of these proposed routing strategies are based on the optimal long-term proportion determined in the previous section. In Section 5, we empirically test several routing strategies using agent data obtained from a large financial service firm's customer service call centers and present results under varying system load conditions. Finally, in Section 6, we provide a summary of the paper along with conclusions and directions for future research.

2 Literature Review

There are several definitions of first call resolution (FCR) in the literature, but in essence it means the proportion of inquires that are successfully addressed after one call. FCR has been getting increasingly more attention in call center management, but our literature survey reveals that most of the interest has come from the practitioners (call center managers, consultants, etc.). We can only find a few research reports on FCR.

For a complete literature review of first call resolution, see Hart *et al.* [13]. They point out the importance of measuring and using FCR. They also point out the existence of different definition of FCR in the call center context. The lack of a standard FCR definition implies that it is not as useful a benchmarking tool as an internal performance measure. The paper states the cost savings that may result from a high FCR (less escalation, less repeat traffic, etc. all leading to lower labor cost), and lists various factors that impact FCR (training, empowerment, technology).

In the literature, there is healthy debate over the merits of measuring satisfaction based on FCR. For example, Read [17] states that surveys reveal that first call resolution drives customer satisfaction, whereas Feinberg *et al.* [8] states that first call resolution (percentage of calls closed on first contact) is not a significant determinant of customer satisfaction in the banking/financial services sector. As the authors admit, due to limitation on data availability, the measure for customer satisfaction (percentage of customers who give "top box" evaluation) is a weak measure and may have accounted for the weakness in the results. Cross [5] warns against using first call

resolution as the only performance measure. He argues that by focusing only on FCR, the manager may overlook opportunities to reduce the volume of non-value-added but simple-to-answer calls. S/he may also overlook opportunities to use call-back or fax-back options to smooth demand.

Early work on routing in call centers considered either a homogenous customer/call population or a homogenous population of servers. Under those conditions, several important results are known about optimal allocation policies or maximal throughput policies under heavy traffic conditions. Most of these use queue backlog rather than waiting time as the control for deciding service allocation. A commonly accepted terminology differentiates between quality driven (QD) and efficiency driven (ED) regimes, emphasizing either utilization of servers or service quality. The balance is described as a Quality and Efficiency driven (QED), which leads to the square root staffing rules. See Halfin and Whitt [12], Borst et al. [4].

More recently several researchers have extended call center routing models to consider a heterogeneous population of service agents. The general field of policies incorporating the service ability of agents is called skills-based-routing. In this context, the maximum feasible arrival rate has been characterized [2, 6, 19], and policies known as maximum pressure or cone policies are known to keep all queues stable whenever that is achievable. These policies is to essentially maximize the inner product (sum of products) of service rate with backlog in the queues - routing calls with large backlogs to servers with high service rates. In [19], these policies are shown to optimize certain backlog-driven performance measures over time.

Another policy proposed in multiskill servers is the Fastest-Servers-First (FSF) policy in [3]. The policy is described as a QED policy with heterogenous servers, and performs better than the homogeneous counterpart for the dynamics described.

The related issue heterogeneous customer value is studied in [10]. The authors analyze the situation where customers differentiated in terms of revenue potential and delay sensitivity. They study staffing, call routing and cross-selling of a heterogeneous customer population, deriving optimal controls. The focus is on how to segment the population itself, and what effect this has on overall profit. Similarly [11] addresses the issue of how many servers are required and how to match them with customers in order to minimize staffing cost, subject to class level QoS constraints. They characterize asymptotically optimal policies as service load grows to infinity. They also show good performance on relatively small systems. Their policy is an idle server based threshold-priority control.

Related to skills based routing is the idea of preference-based routing presented in [18]. Tra-

ditional skills based routing algorithms try to match call types with agent skills subject to service constraints. They do not consider agents' preferences for call types. Given the high turnover (churn) at call centers and the associated high costs, they propose routing algorithms that account for agent skills and preferences. They do so by assigning values to call-agent combinations that incorporate management's judgment of the value of such pairings and each agent's preferences for the call types. Moreover, by letting the values be based on call resolution rates, this framework also applies to the problem we consider in this paper.

More closely related to our study here is [7]. This paper considers call resolution probabilities in making call routing decisions. There is only one call type, but many agent classes. The agent classes are differentiated by their call handle time (service rate) and call resolution probabilities. They show that agent classes can be ranked by their call resolution rate (call resolution probability times service rate), the so-called $p\mu$ policy. To minimize the average total time to resolve a call, they show that there is always a preferred agent class, the one with the highest $p\mu$, to route the calls to, and when all agents in that class are busy, it is optimal to route to other classes following a state-dependent threshold policy. Using numerical tests, the authors show that a routing policy that overlooks the call resolution probability differences can perform poorly, which illustrates the importance of routing based on call resolution. To simplify the routing policy, they show numerically that the optimal state-independent policy already captures almost all the benefits of the state-dependent threshold policy. Moreover, routing solely based on the $p\mu$ index, without the use of thresholds, allows the call center to get most of the benefits.

3 Model and Problem Formulation

A customer's experience during a service encounter consists of two parts: the wait and the service itself. The wait-related measures such as ASA deal with the first aspect while resolution related measures such as CR deal with the second. Given the heterogeneity of the agents – some may be faster while others may yield better customer resolution – oftentimes there is an inherent tradeoff in the routing decisions. If the aim is to reduce overall wait, then the system should route calls to agents in such a way to maximize the effective rate (accounting for re-service of those un-resolved calls) at which calls leave the system. If the aim is to increase resolution, then the system should route calls to agents in such a way to maximize the overall CR.

To achieve the first, one would think it best to route calls to agents who can handle it the

fastest, sometimes even withholding a call in queue to wait for that agent to free up, but that doesn't account for the re-service of calls that were not resolved. A more effective way is to route calls to the agents with the highest "resolution rate" which is the product of service rate and CR (see de Véricourt and Zhou [7] for details). To achieve the second, it is clearly optimal to route each call type to the agent group who can handle it the best (highest CR rate), sometimes even withholding a call in queue to wait for that agent to free up. However, this may put undue burden on some agent groups (even overloading them) while some other agent groups may become idle – a very inefficient use of resources.

In this paper we aim to find routing policies that achieve a balance between the two goals of short wait and high resolution. One policy we will test comes from de Véricourt and Zhou [7]: routing calls to available agents with the highest resolution rate, or the so called "$p\mu$ rule". On the "wait-resolution" spectrum, this policy resides close to the "wait" end because it is derived with the aim to reduce overall wait time, accounting for re-service of resolved calls. At the other end of the spectrum, we will derive a policy that aims to maximize CR but with constraints on minimum and maximum utilization targets for each of the agent groups and each call type. Then we propose a policy that "continualize" the spectrum. With different parameter values it can move between the two ends of the spectrum. This way, a call center manager can pick the parameter to achieve the wait-resolution combination that is right for his/her call center.

Next, we present a model that calculates the maximal call resolution rate that can be achieved by any stationary Markovian policy given agent utilization bounds. This gives us a sense of how far the "resolution" end of the "wait-resolution" spectrum stretches out. After that we will present a model that analyzes the naive first-come-first-served policy, to serve as a benchmark of how much benefit can be gained. In Section 3.3, we will describe the policies we test in the numerical analysis that are based on the $p\mu$ rule in de Véricourt and Zhou [7], the x_{ij}s derived in Section 3.1, the FCFS policy in Section 3.2.

3.1 Max-CR Problem Formulation

Here we formulate the problem of maximizing the overall expected CR rate, subject to minimum and maximum utilization targets for each of the agent groups and each call type. We assume that for each agent group, performance parameters are known.

Let $i = 1, 2, ...I$ index the different queues and $j = 1, 2, ...J$ index the different groups of agents, with n_j agents in group j. Clearly routing decision must be made dependent on the state, as

calls come in and as the agents become available. But any stationary Markovian policy will result in a continuous time Markov chain (CTMC). We can analyze the CTMC to find the stationary distribution of the system in order to study the system's performance. For a CTMC that's the result of a stationary Markovian policy, there is a corresponding set of variables x_{ij} that represent the percentage of calls from queue i to be handled by agents from class j in the steady state. This is all we need to figure out the CR that the policy will achieve. Therefore, the x_{ij}s are our decision variables. We will formulate a mathematical program below to find the x_{ij} that maximizes the CR subject to utilization constraints. To ascertain the feasibility of such x_{ij} we only need to note that a naive policy of routing a type i call to type j agent with probability x_{ij} (and letting the call stay there even all type-j agents are busy and other agents are idle) achieves this x_{ij}, and thus the maximum CR.

We formalize the optimization model below, but first consider the *effective* arrival rate to each queue, taking into account callbacks due to unresolved earlier calls. We assume for this study that customers have no alternative to resolving their call through the call center, and hence all unresolved calls will return as future arrivals. For each agent group j, they serve type-i calls at rate μ_{ij} and successfully resolve each call with probability p_{ij}. Denote λ_i to be the arrival rate of first time type-i customers. The effective arrival rate $\bar{\lambda}_i$, accounting for all the re-services, explicitly depends on the choice of the x_{ij} values, as the x_{ij} values determine the percentage of customers who call back. In particular, we have:

$$\bar{\lambda}_i = \lambda_i + \lambda_i(\sum_j (1-p_{ij})x_{ij}) + \lambda_i(\sum_j (1-p_{ij})x_{ij})^2 + \lambda_i(\sum_j (1-p_{ij})x_{ij})^3 \dots$$

The kth term on the right hand side of the equation corresponds to the expected number of customer who make a total of k calls before getting resolved, $k = 2, 3, ...$ Now since $\sum_j (1-p_{ij})x_{ij} < 1$, we have:

$$\bar{\lambda}_i = \frac{\lambda_i}{1 - \sum_j (1-p_{ij})x_{ij}} \tag{1}$$

For an agent of type j, their total arrival rate for jobs of type i is $\bar{\lambda}_{ij} = \frac{1}{n_j}\bar{\lambda}_i x_{ij}$, and hence their total utilization is $\sum_i \frac{\bar{\lambda}_{ij}}{\mu_{ij}} = \sum_i \frac{\bar{\lambda}_i x_{ij}}{n_j \mu_{ij}}$. For our routing assignments, we require that each agent in group j be utilized between a lower bound ρ_j^- and upper bound ρ_j^+.

To maximize the overall CR rate, we formulate the following optimization problem:

maximize $\sum_{i,j} \bar{\lambda}_i p_{ij} x_{ij}$ (max total rate of resolution)

subject to

$$\begin{array}{rcccll} 0 & \leq & x_{ij} & \leq & 1 \quad \forall i,j & \text{(fraction of calls bound)} \\ \sum_j x_{ij} & = & 1 & & \forall i & \text{(total calls routed to different agent groups)} \\ \rho_j^- & \leq & \sum_i \frac{\bar{\lambda}_i x_{ij}}{n_j \mu_{ij}} & \leq & \rho_j^+ \quad \forall j & \text{(utilization of each agent)} \end{array}$$

If we substituting $\bar{\lambda}_i$ by (1), then we change the objective function and the agent utilization constraint, respectively, to:

$$\max_{x_{ij}} \quad \sum_{i,j} \frac{\lambda_i p_{ij} x_{ij}}{1 - \sum_j (1 - p_{ij}) x_{ij}} \tag{2}$$

and

$$\rho_j^- \leq \sum_i \frac{(\lambda_i / n_j \mu_{ij})\, x_{ij}}{1 - \sum_j (1 - p_{ij}) x_{ij}} \leq \rho_j^+ \qquad \forall j. \tag{3}$$

Both are quadratic so the problem can be efficiently solved with any good commercial solver.

Note that this formulation has the advantage of allowing a call center to solve the closely related problem of maximizing *first* call resolution (FCR), by simply replacing the above objective function with $\sum_{i,j} \lambda_i p_{ij} x_{ij}$ (which also makes the objective function linear). In a traditional Markov Decision Process (MDP) based approach, this is hard to do because one needs to keep separate track of new arrivals and returned jobs. This enlarges the state space, oftentimes yielding the MDP problem intractable. With this optimization approach, we can easily find the x_{ij}s that optimize FCR and then seek policies that achieve (or approach) the optimal x_{ij}s, as we will do in Section 3.3.

It is possible that the solution to the optimization problem above may unfairly affect some job types more than others. One way to protect against this is to constrain the effective utilization of each job type i. That is, each job type i must be served at total utilization between τ_i^- and τ_i^+. To do that, we must first define what we mean by utilization of call type i. This can be done by calculating the effective service attention from all agent types.

For an agent of type j, their total fraction of time spent serving queue i is $\frac{\frac{\bar{\lambda}_{ij}}{\mu_{ij}}}{\sum_{i'} \frac{\bar{\lambda}_{i'j}}{\mu_{i'j}}}$. Therefore the total service rate to jobs of type i is $\bar{\mu}_i = \sum_{j=1}^{J} n_j \mu_{ij} \frac{\frac{\bar{\lambda}_{ij}}{\mu_{ij}}}{\sum_{i'} \frac{\bar{\lambda}_{i'j}}{\mu_{i'j}}}$. The total effective utilization of queue i is then seen to be

$$\frac{\bar{\lambda}_i}{\bar{\mu}_i} = \frac{\bar{\lambda}_i}{\sum_{j=1}^{J} \frac{n_j \bar{\lambda}_{ij}}{\sum_{i'} \frac{\bar{\lambda}_{i'j}}{\mu_{i'j}}}} = \frac{\bar{\lambda}_i}{\sum_{j=1}^{J} \frac{\bar{\lambda}_i x_{ij}}{\sum_{i'} \frac{\bar{\lambda}_{i'} x_{i'j}}{n_j \mu_{i'j}}}} = \frac{1}{\sum_{j=1}^{J} \frac{n_j x_{ij}}{\sum_{i'} \frac{\bar{\lambda}_{i'} x_{i'j}}{\mu_{i'j}}}} \tag{4}$$

Now, given upper and lower bounds on utilization per queue, we can write:

$$\tau_i^- \leq \frac{1}{\sum_{j=1}^{J} \frac{n_j x_{ij}}{\sum_{i'} \frac{\bar{\lambda}_{i'} x_{i'j}}{\mu_{i'j}}}} \leq \tau_i^+, \qquad \forall i. \tag{5}$$

The right hand inequality, for example, can be rewritten as

$$\sum_{j=1}^{J} \frac{n_j x_{ij}}{\sum_{i'} \frac{\bar{\lambda}_{i'} x_{i'j}}{\mu_{i'j}}} \geq \frac{1}{\tau_i^+} \tag{6}$$

Again, this is a quadratic constraint and can be easily handled by good solvers.

3.2 First Come First Served

As a benchmark, we will study the First Come First Served (FCFS) policy. In this case, different types of calls arrive to the same queue and are taken FCFS. If a call arrives to find several agents available, it is randomly assigned to one of them. Otherwise, it is queued. When agents become idle, they take from the head of the queue. In such a system, each agent will have the same utilization in the long run. Therefore, we must have

$$\sum_i \frac{x_{ij} \bar{\lambda}_i}{n_j \mu_{ij}} = \sum_i \frac{x_{ik} \bar{\lambda}_i}{n_k \mu_{ik}}, \qquad \forall j, k. \tag{7}$$

Also, we have

$$\sum_j x_{ij} = 1, \qquad \forall i. \tag{8}$$

Given that calls are randomly assigned, the call type distribution among all the calls taken by different agent classes should be the same. So we must also have:

$$\frac{x_{ij} \bar{\lambda}_i}{\sum_{i'} x_{i'j} \bar{\lambda}_{i'}} = \frac{x_{ik} \bar{\lambda}_i}{\sum_{i'} x_{i'k} \bar{\lambda}_{i'}}, \qquad \forall i, j, k. \tag{9}$$

These equations should uniquely determine the solution. However, it's too complicated and not as intuitive as the alternative approach we give below: Let there be only one type of call, with total arrival rate of $\sum_i \bar{\lambda}_i$. If a call arrives to find several agents available, it is randomly assigned to one of them. Otherwise, it is queued. Calls are queued and taken FCFS. When agents become

idle, they take from the head of the queue. Moreover, when a call is taken by a type-j agent, its service time is exponential with rate μ_{ij} with probability $\frac{\bar{\lambda}_i}{\sum_{i'} \bar{\lambda}_{i'}}$.

Let z_j be the percentage of calls that are taken by agent type j. Then, we must have $\sum_j z_j = 1$. Moreover, because of the FCFS and random assignment policy, we must have equal utilization among all the agent types. For type j the effective arrival rate is $z_j \left(\sum_i \bar{\lambda}_i\right)$, and the average service time is $\sum_i \frac{\frac{\bar{\lambda}_i}{\sum_{i'} \bar{\lambda}_{i'}}}{\mu_{ij}}$. Therefore,

$$\frac{z_j \left(\sum_i \bar{\lambda}_i\right) \sum_i \frac{\frac{\bar{\lambda}_i}{\sum_{i'} \bar{\lambda}_{i'}}}{\mu_{ij}}}{n_j} = \frac{z_k \left(\sum_i \bar{\lambda}_i\right) \sum_i \frac{\frac{\bar{\lambda}_i}{\sum_{i'} \bar{\lambda}_{i'}}}{\mu_{ik}}}{n_k} \qquad \forall j, k \tag{10}$$

which simplifies to

$$z_j \frac{\sum_i \left(\frac{\bar{\lambda}_i}{\mu_{ij}}\right)}{n_j} = z_k \frac{\sum_i \left(\frac{\bar{\lambda}_i}{\mu_{ik}}\right)}{n_k} \qquad \forall j, k. \tag{11}$$

Along with $\sum_j z_j = 1$, these equations uniquely determine all the z_js. Then the x_{ij}s can be derived as follows:

$$x_{ij} = z_j, \qquad \forall i. \tag{12}$$

Now, note that the $\bar{\lambda}_i$s are determined by the x_{ij}s, so we still need to solve a system of equations to determine all the x_{ij} and $\bar{\lambda}_i$ jointly.

3.3 Routing Policies

If we denote the target x_{ij} proportions derived in Section 3.1 by x^*_{ij}, then two simple rules will guarantee the x^*_{ij}s are used. These rules are randomized and round robin routing, with exactly x^*_{ij} of calls of type i sent to agent group j. This can be achieved by a coin-flip operation (randomized) or a set schedule with the appropriate proportions used. These are non-dynamic in that no backlog information is taken into account, so one would expect them to be relatively poor in terms of wait time performance.

Several natural routing rules do consider x^*_{ij} but route according to other system conditions such as backlog and utilization levels. Given fixed arrival and service rates, these will lead to long term $\hat{x}_{ij}$ values, and we are interested to see how these values, and particularly the corresponding objective values compare to the optimal solution.

Unfortunately general closed form expressions for wait time are impossible to calculate for most of these rules, but the existence of long term averages is guaranteed by the Markovian dynamics of the system. Hence estimations from simulation can give a great deal of insight into the relative performance levels of various routing algorithms.

Let $Q_i(t)$ be the number of waiting calls of type i at time t.

The following routing rules were tested in our experiments:

1. **FCFS** When agent j becomes free, select the call that has been waiting the longest.

2. **minAAHT**: When agent j becomes free, select $\arg\max_{i:Q_i(t)>0}\{p_{ij}\mu_{ij}\}$. This selects the call type where the agent has the highest effective service rate.

3. **minDiffAHT**: When agent j becomes free, select $\arg\max_{i:Q_i(t)>0}\{p_{ij}\mu_{ij} - \max_{k\neq j} p_{ik}\mu_{ik}\}$. This selects the call type where the agent has the highest *relatively* effective service rate.

4. **maxCR** When agent j becomes free, select $\arg\max_{i:Q_i(t)>0}\{p_{ij}\}$. This selects the call type where the agent is *most likely to resolve*.

5. **maxDiffCR**: When agent j becomes free, select $\arg\max_{i:Q_i(t)>0}\{p_{ij} - \max_{k\neq j} p_{ik}\}$. This selects the call type where the agent is *relatively most likely to resolve*.

6. **OptXRand** Upon arrival, each call that arrives is assigned to agent j with probability x^*_{ij} values. Calls wait in agent-specific queues.

7. **OptMaxDev** Let $\hat{x}_{ij}(t)$ be the proportion of calls of type i that have been handled by agents in group j up to time t.

   (a) When there is more than one agent waiting when a call of type i arrives, select agent from the group with the maximum $\arg\max_{j:j \text{ free}}\{x^*_{ij} - \hat{x}_{ij}(t)\}$. Here we are choosing the agent who is *farthest behind* on calls of type i relative to the optimal values.

   (b) When an agent from group j comes free and there is more than one type of call waiting, select the call from $\arg\max_{i:Q_i(t)>0} x^*_{ij} - \hat{x}_{ij}(t)$. Here the agent is selecting the call for which he/she is *farthest behind* relative to the optimal values

   Note that if all agents are under-utilized (and hence usually available on call arrival), then this would perform similar to a round-robin policy.

8. **OptMaxCallDev** This is the same as Policy **OptMaxDev** except that it does not allow for calls to be chosen by agents.

9. **CallSwap1** Calls are routed to agent groups according to Policy **OptXRand**. When an agent comes free, the queue for that agent group is checked to see if it is empty. If not, then

take the call at the head of the queue. If the queue is empty, then all queues are checked to see if any are full (defined as more than a certain number of calls waiting). If more than one evaluates to "full", then the first call in queue for the first queue evaluating to full is removed and reassigned to the empty queue. If no queue is "full", then no swapping.

10. **CallSwap2** Calls are routed to agent groups according to Policy **OptXRand**. But agents are allowed to swap calls up to an upper limit. When an agent comes free, the queue for that agent group is checked to see if it is empty. If not, then take the call at the head of the queue. If the queue is empty, then check the other agent queues to "take back" any calls that have been "lent" before. If there are no such calls, then check the non-empty queues to "borrow" calls from other agent groups, as long as the upper limit has not been reached. Update the "borrow-lend" list accordingly.

For benchmark purposes we start with the Policy **FCFS**. This is greedy in the short term with respect to ASA, so should perform reasonably well on that dimension, and is also one of the easiest policies to implement in practice.

Policies **minAAHT** and **minDiffAHT** are motivated by de Véricourt and Zhou [7]. Calls are routed to agents who have the highest absolute and relative resolution rate, or the $p\mu$ index, respectively. We expect this to do very well in terms of the ASA performance metric, because the effectiveness of $p\mu$ rule is established under the objective of overall minimum wait. It is not obvious how it should perform on the CR dimension.

Policies **maxCR** and **maxDiffCR** are greedy and myopic. They aim to route calls to the agent who can has the highest resolution rate for this call type. However, it does not account for the service rate. If an agent group works very slow but has high resolution rates, then it will be heavily loaded. That the agents are slow clearly will result in long waits. So we expect these two policies to perform poorly on the ASA metric. In terms of the CR metric, because of the myopia of the policy (it routes to the agent with the highest p index *at the moment*, we expect the CR performance to be worse than that of Policy **OptXRand**, which should maximize the overall total CR rate. Under Policy **OptXRand**, once a type-i call is routed according to x^*_{ij} it stays in the queue specific to agent group j and become inaccessible to other agents. Thus, it completely loses the pooling effect so important in large call centers. Consequently, while Policy **OptXRand** should perform the best on the CR metric, one would expect it to perform poorly on the ASA metric.

On the ASA-CR performance spectrum, Policies **minAAHT** and **OptXRand** lie on the two

ends. A call center manager may want to find a policy that locates somewhere between the two. The rest of the policies aim to do that.

Policy **OptMaxDev** aims to mitigate the loss of pooling effect under Policy **OptXRand** by not idling some agent groups while other agent groups are swamped. All the agents share the same pool of calls, but when calls are routed, the aim is to minimize the deviation of realized x_{ij} from the x^*_{ij}. The policy specifies the rules to use when agents become idle (which call to take next) and when calls arrive (which agent to route to). Policy **OptMaxCallDev** is a slight variation of Policy **OptMaxDev**. It simplifies Policy **OptMaxDev** by allowing the selection only when calls arrive (that is, calls can select agents, but not vice versa).

Policies **CallSwap1** and **CallSwap2** also aims to improve Policy **OptXRand** by trying to stay close to the target x^*_{ij} but at the same time allowing pooling across call types and agent types. Policy **CallSwap1** does that by allowing agents to "borrow" calls from each other (this restores the pooling effect). However, to prevent the realized x_{ij} straying too far from the target x^*_{ij}, agents can only borrow from another agent group only if that group has more than a threshold number calls waiting in queue. Clearly the lower this threshold, the more pooling is restored (when threshold is zero, there is complete pooling), but at the same time the farther the realized x_{ij}s stray from the target x^*_{ij}s. Similarly, Policy **CallSwap2** allows agents to "borrow" calls from each other to restore the pooling effect. Similarly, to stay close to the target x^*_{ij}, the number of calls each agent group can borrow is limited by a fixed number. Clearly, the higher this threshold, the more pooling is restored, but at the same time the farther the realized x_{ij}s stray from the target x^*_{ij}s. In the limit, because the upper limit is fixed, the realized long-run average proportions x_{ij}s are the same as x^*_{ij}. As we have already pointed out, one advantage of these two policies is that they each have a parameter (the threshold) that can be varied so that call center can achieve performances close to either end of the ASA-CR spectrum. The choice of the threshold in both policy is therefore essential in determining the balance of the ASA-CR tradeoff.

4 Numerical Analysis

The optimal x^*_{ij} routing probabilities for policy **OptXRand** can be calculated using the mathematical program in Section 3.1, from which we can calculate the resultant CR. Moreover, the CR for policy **FCFS** can be calculated using the procedure in Section 3.2. The CR for the rest of the policies in Sections 3.3, as well as the ASA for all the policies, are analytically intractable.

Agent Type	Call Type	CR (%)	AHT (seconds)
1	1	92.94	230.76
2	1	93.69	229.10
3	1	91.84	263.33
4	1	93.16	231.56
1	2	95.15	232.17
2	2	96.50	322.27
3	2	95.37	246.16
4	2	95.40	238.10
1	3	97.43	138.94
2	3	97.61	166.19
3	3	97.23	170.11
4	3	97.80	160.62
1	4	98.68	192.02
2	4	98.76	177.23
3	4	99.13	182.39
4	4	99.01	199.31

Table 1: Service and Resolution Rates

To evaluate their performance for comparison, we conduct extensive simulation tests. Below we describe the parameters first, then we talk about the simulation setup. Finally we present and discuss the simulation results.

We assume four agent groups. And in three sets of tests, we let the number of agents per group be 15, 60, and 120 respectively. There are also four job types. The arrival rates for the first set of experiments (15 agents/group) are (4.02, 3.24, 2.63, 2.11) jobs per minute for the four call types respectively. We multiply these rates by 4 and 8 respectively to get the arrival rates for the second and third sets of tests. Table 1 lists call type - agent type specific resolution rate (CR) and average service time (AHT). Note that these service and resolution rates come from a real call center.

To simulate each policy, we use the events corresponding to the first 3 hours in real time as warm up. After that, each replication length corresponds to one hour in real time. For the three sets of simulations, the number of replications are 450, 125, and 65 respectively. Once all the simulations are completed, we collect output statistics and then calculate the grand average over all the replications and all the sets to find steady-state performance of all the policies. We also calculate the standard deviations of the performances. The results are summarized in Figure 1.

The results in Figure 1 confirm some of our intuitions, and they provide additional insights:

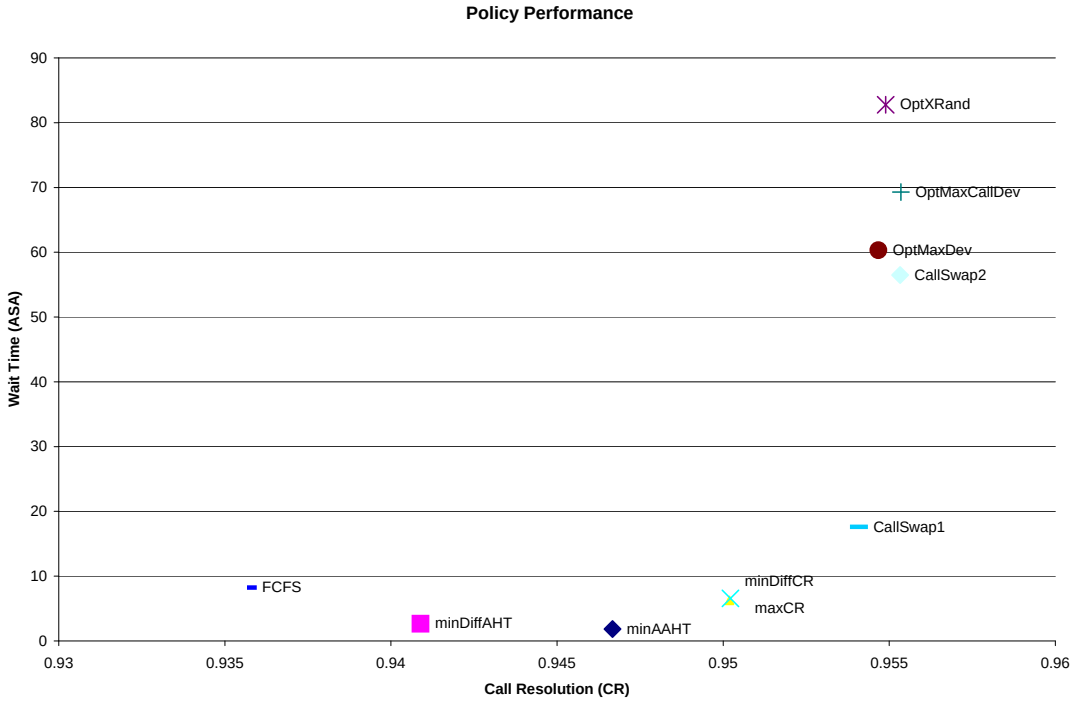

Figure 1: FCR vs. average speed of answer for various rules.

- The "efficient frontier" lies in the lower right-hand corner of the graph (high CR, low ASA), and the FCFS, as expected is not on the frontier. It is clearly dominated by Policy **minAAHT**. Also Policy **minDiffAHT**, which uses the *relative* $p\mu$ index in routing, is dominated by Policy **minAAHT**), which uses the *absolute* $p\mu$ index.

- Policy **minAAHT** does represent one end of the ASA-CR spectrum (which is the "efficient frontier" on Figure 1) – it has extremely low ASA, as expected, but medium CR.

- Policy **OptXRand** also seems to represent the other end of the ASA-CR spectrum (it has the second highest CR and the small differences can be attributed to the simulation error).

- All the other Policies lie between the two ends of the spectrum. But Policy **CallSwap2**, which uses a threshold of 5 to make sure that realized x_{ij} approaches x^*_{ij} in the long run), seems to dominate Policies **OptMaxCallDev** and **OptMaxDev**, which only uses x^*_{ij} as guidance in call routing.

- Policies **maxCR** and **maxDiffCR**, which are based on just the p_{ij} parameters, result in lower CR than Policies **CallSwap1** and **CallSwap2** which are based on x^*_{ij}. This is reassuring because we know Policies **maxCR** and **maxDiffCR** have full pooling while Policies **CallSwap1** and **CallSwap2** do not completely take advantage of pooling. On the other hand, it reveals that routing to maximize CR in a greedy and myopic fashion (as Policies **maxCR** and **maxDiffCR** do) does not work as well as Policies **CallSwap1** and **CallSwap2**, which maximize CR in the long run, accounting for all the re-services.

- We believe that by varying the threshold value in Policies **CallSwap1** and **CallSwap2** we can move their corresponding points on the graph along the frontier. We are conducting further studies to see this effect.

5 Conclusions

Call resolution has received great attention in call center management, but there are very little analytical models and insights to guide call center managers in practice. Our paper is among the first to present an analytical model where call resolution not only is modeled but also plays an critical role in deciding the optimal policy and the resultant performance in terms of wait time (ASA) and customer satisfaction (CR). The insights we generated through analytical modeling and

simulation are useful to call center managers in analyzing the ASA-CR tradeoff and deciding where on the efficient frontier to be, and how to get there. The policies we propose are intuitive, based on sound scientific analysis, and implementable. We think this is just the first step in a very promising research direction.

References

[1] Z. Aksin, M. Armony, and V. Mehrotra. The modern call-center: A multi-disciplinary perspective on operations management research. *Working Paper*, 2007. Available at http://www.stern.nyu.edu/om/faculty/armony/research/CallCenterSurvey.pdf.

[2] M. Armony and N. Bambos. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems*, 44(3):209, 2003.

[3] Mor Armony. Dynamic routing in large-scale service systems with heterogeneous servers. *Queueing Systems*, 51:287–329, 2005.

[4] S. Borst, A. Mandelbaum, and M.I. Reiman. Dimensioning large call centers. *Operations Research*, 2004.

[5] Kelvin F Cross. Call resolution: The wrong focus for service quality? *Quality Progress*, 33(2):64–67, February 2000.

[6] J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2):197–218, 2005.

[7] Francis de Vèricourt and Yong-Pin Zhou. A routing problem for call centers with customer callbacks after service failure. *Operations Research*, 53(6), Nov-Dec 2005.

[8] Richard A Feinberg, Leigh Hokama, Rajesh Kadan, and IkSuk Kim. Operational determinants of caller satisfaction in the banking/financial services call center. *The International Journal of Bank Marketing*, 20(4/5):174–180, 2002.

[9] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5:79141, 2003.

[10] I. Gurvich, M. Armony, and C. Maglaras. Cross-selling in a call center with a heterogeneous customer population. *White Paper*, 2006.

[11] I. Gurvich, M. Armony, and A. Mandelbaum. Service level differentiation in call centers with fully flexible servers. *Management Science*, To Appear.

[12] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 1981.

[13] Mike Hart, Bastien Fichtner, Espen Fjalestad, and Steven Langley. Contact centre performance: In pursuit of first call resolution. *Management Dynamics*, 15(4):17–28, 2006.

[14] Pierre L'Ecuyer. Modeling and optimization problems in contact centers. *Third International Conference on the Quantitative Evaluation of Systems - (QEST'06)*, pages 145–156, 2006.

[15] A. Mandelbaum and S. Zeltyn. Service engineering in action: The palm/erlang-a queue, with applications to call centers. In Spath D. and K.-P. Fhnrich, editors, *Advances in Services Innovations*, pages 17–48. Springer-Verlag, 2007.

[16] E. Pinker and R. Shumsky. The efficiency-quality trade-off of cross-trained workers. *Manufacturing and Service Operations Management*, 2(1):32–48, Winter 2000.

[17] Brendan B. Read. Call center checkup. *Call Center Magazine*, June 2003.

[18] Michael E. Sisselman and Ward Whitt. Value-based routing and preference-based routing in customer contact centers. *Production and Operations Management*, Forthcoming.

[19] A. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1–53, 2004.