Cross Training in Call Centers with Uncertain Arrivals and Global Service Level Agreements

Volume 16, *Number* 3
September 2010*, pp*. 307-329

Thomas R. Robbins
East Carolina University
Greenville, NC
(robbinst@ecu.edu)

D. J. Medeiros, Terry P. Harrison
Pennsylvania State University
University Park, PA
(djm3@psu.edu, tharrison@psu.edu)

We consider cross-training in inbound call centers with non-stationary, uncertain arrival rates and global Service Level Agreements. We investigate the option of cross training a subset of agents so that they may serve calls from two separate queues, a process we refer to as partial pooling. We develop a simulation-based search heuristic that finds near-optimal schedules for a pool of two different queue types. We analyze the benefits of partial pooling and characterize the conditions under which pooling is most beneficial. We find that cross training a modest portion of the staff yields significant benefits even when cross training is costly.

Keywords: Call Center Scheduling, Simulation, Cross-Training, Simulation-Based Optimization

1. Introduction

A call center is a facility designed to support the delivery of some interactive service via telephone communications, typically an office space with multiple workstations manned by agents who place and receive calls (Gans, Koole *et al.* 2003). Call center applications include telemarketing, customer service, help desk support, and emergency dispatch. Call centers are a large and growing component of the U.S. and world economy; the United States was estimated to employ 2.1 million call center agents by 2008 (Aksin, Armony *et al.* 2007). Large-scale call centers are technically and managerially sophisticated operations and have been the subject of substantial academic research.

Staffing is a critical issue in call center management as direct labor costs often account for 60-80% of the total operating budget (Aksin, Armony *et al.* 2007). This paper addresses the staffing problem in a call center with highly variable and uncertain arrival rates. Given two call types, each with a service level agreement, we seek to satisfy the service level objectives with high probability while minimizing overall staffing costs. Agents are normally dedicated to a single call type, but a subset may be cross-trained (at considerable expense) to answer calls of both call types.

A similar problem is found in Wallace and Whitt (2005) which admits multiple call types, with every agent trained to handle a fixed number of those types. In contrast,

our scenario has two call types and partial cross-training; 100% cross training is uneconomical. Wallace and Whitt find that training every agent in two skills provides the bulk of the benefit, while additional training has a relatively low payoff. We seek to find the appropriate number of agents to cross-train while explicitly considering the associated incremental costs. Wallace and Whitt (2005) examine cross training in steady state, where arrival rates and staff levels are fixed. We focus on the case where both arrival rates and staff levels change dramatically over time. With changing arrival rates, customer abandonment (losing patience while on hold and disconnecting the call) becomes an important issue.

This paper is motivated by work with a provider of outsourced technical support delivered via globally distributed call centers. The bulk of their business, and the focus of our research, is an inbound call center operation which provides help desk support to large corporate and government entities. While the scope of services varies from account to account, many accounts are 24 x 7 support and virtually all accounts are subject to some form of Service Level Agreement (SLA). There are multiple types of SLAs, but the most common specifies a minimum level of the Telephone Service Factor (TSF). A TSF SLA specifies the proportion of calls received that must be answered within a given time. For example, an 80/120 SLA specifies that 80% of calls received must be answered within 120 seconds. An important point is that the SLA applies to an extended period, typically a week or a month. Thus, a help desk is often staffed so the service level is sometimes underachieved, sometimes overachieved, and is on target for the week or month.

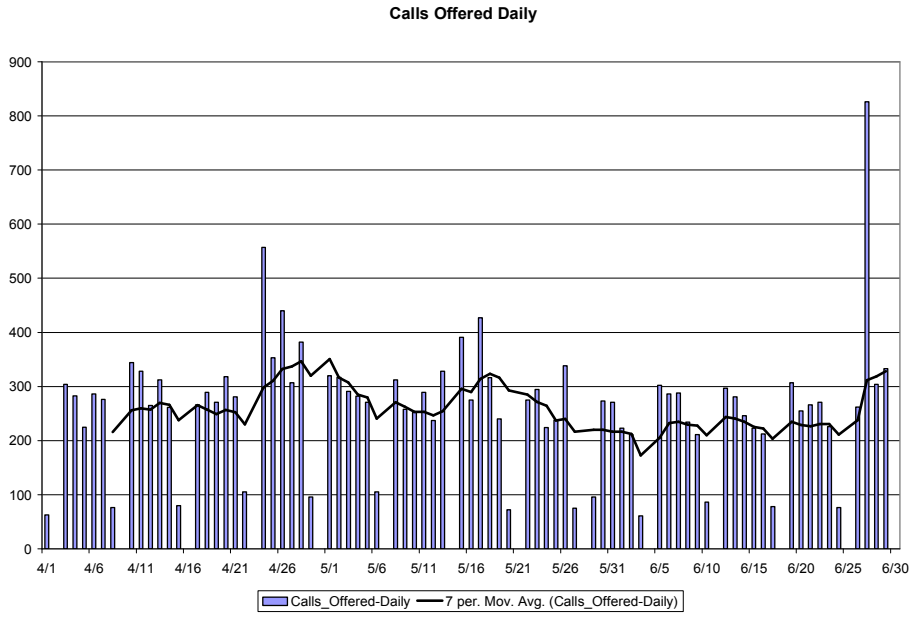

Figure 1 *Sample Daily Arrival Pattern*

The key challenge involved with staffing this call center is meeting a fixed SLA with a variable and uncertain arrival rate pattern. The number of calls presented in

any half hour period is highly variable with multiple sources of uncertainty. Figures 1 and 2 show daily call volume over a three-month period along with the range of hourly call volume over an 8-week period. These are representative samples from actual projects supported by this provider.

Both graphs show strong periodic variation in call arrivals. Mondays tend to be the highest volume days with volumes decreasing over the course of the week. Call volume on weekends is a small fraction of the weekday volume (the desk shown in Figure 1 is closed on Sundays). Within a day, there are two demand peaks separated by a lunch break. Both graphs also reveal significant stochastic variability between and within days. In Figure 2 the inner region represents the minimum volume presented in each period, the overall envelope is the maximum volume presented, and the difference reflects the variability for the period.

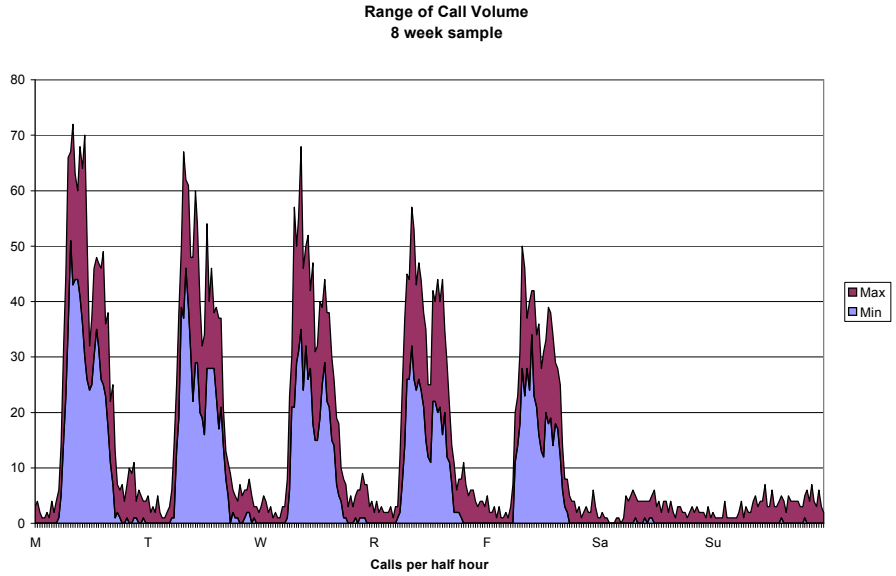

Figure 2 *Range of Call Volumes*

The staffing challenge in this call center is to find a minimal cost staffing plan that achieves a global service level target with a high probability. The staffing plan must obviously be locked in before arrival rate uncertainty is revealed.

In this paper we apply a simulation based optimization heuristic to create near-optimal call center agent staff plans that utilize partial pooling in the presence of random arrival rates to several "real world" cases. We analyze these staff plans to quantify the economic benefits of partial pooling and to characterize the conditions under which this partial pooling is most beneficial.

In Section 2 we briefly review the relevant call center literature. Section 3 presents the basic call center configuration and cross training model examined in the rest of the paper along with a formal statement of the staffing optimization model. In Section 4 we outline our simulation-based search heuristic used to solve the problem. In Section 5 we present the results of numerical experiments that examine the use of

partial pooling in a realistic setting, applying the search algorithm to schedule sample call centers. Section 6 provides extension and future directions for this research.

2. Literature Survey

Call centers have been the focus of significant academic research. A detailed review of the call center literature is provided in Gans, Koole *et al.* (2003). More recent work is summarized in Aksin, Armony *et al.* (2007). Empirical analysis of call center data is provided in Brown, Gans *et al.* (2005).

Call centers are often analyzed using queuing theory, often as an M/M/n (Erlang-C) queuing system. In many call center applications a non-negligible proportion of callers hang up (abandon) the queue prior to being served. Under these circumstances the Erlang-A model may be used since it models abandonment. The details of the Erlang-A model are provided in Mandelbaum and Zeltyn (2004). Methods for approximating Erlang-A results are described in Garnett, Mandelbaum *et al.* (2002). An assessment of the Erlang-A model to parameter sensitivity is provided in Whitt (2006a).

Customer service is an important consideration in call centers, and many centers are subject to SLAs. Milner and Olsen (2008) examine contract structures in call centers with service level agreements. Baron and Milner (2006) examine optimal staffing levels under various SLAs. These papers classify SLAs as Individual-based (IB), Period-based (PB), or Horizon-based (HB). IB-SLAs assess a financial penalty for every customer not served within the specified service level. The PB-SLA specifies penalties for each time period in which the service level target is not achieved. The HB-SLA specifies penalties for service level shortcomings over an extended period such as a week or month. In this paper we examine scenarios where an HB-SLA has been specified with the horizon specified as one week.

Most call center scheduling models in the literature implement a hard constraint for service level on a period by period basis, *i.e.* a PB-SLA, but a few models are formulated to solve a global service level requirement, *i.e.* an HB-SLA. It is our experience that outsourcing contracts often specify an HB-SLA and all of the projects we examined were subject to this type of SLA. Koole and van der Sluis (2003) attempt to develop a staffing model that optimizes a global objective based on an HB-SLA using a local search algorithm. They require agent schedules with no breaks, and assume no abandonment. Their model also assumes a time varying, but known arrival rate. Cezik and L'Ecuyer (2008) solve an HB-SLA problem using simulation and integer programming; they use simulation to estimate service levels then solve the scheduling problem using integer programming. Their model is an extension of the model presented in Atlason, Epelman *et al.* (2004). In a related paper Avramidis, Chan *et al.* (2007) use a local search algorithm to solve the same problem. A related model is presented in Avramidis, Gendreau *et al.* (2007). Fukunaga, Hamilton *et al.* (2002) describe a commercial scheduling application widely used for call center scheduling. Global service level targets are modeled as soft constraints while certain staffing restrictions are modeled as hard constraints. The algorithm uses an artificial intelligence-based search heuristic. Atlason, Epelman *et al.* (2008) develop an algorithm that combines server sizing and staff scheduling into a single optimization problem. This model explicitly considers the impact of staffing in one time period on performance in the subsequent period. The algorithm

utilizes discrete event simulation to calculate service levels under candidate staffing models, and a discrete cutting plane algorithm to search for improving solutions.

The issue of arrival rate uncertainty has been addressed in several recent papers. Both major call center reviews (Gans, Koole *et al.* 2003; Aksin, Armony *et al.* 2007) have sections devoted to arrival rate uncertainty. Brown, Gans *et al.* (2005) perform a detailed empirical analysis of call center data. While they find that a time-inhomogeneous Poisson process fits their data, they also find that arrival rate is difficult to predict and suggest that the arrival rate should be modeled as a stochastic process. Many authors argue that call center arrivals follow a doubly stochastic process, a Poisson process where the arrival rate is itself a random variable (Chen and Henderson 2001; Whitt 2006b; Aksin, Armony *et al.* 2007). Arrival rate uncertainty may exist for multiple reasons. Arrivals may exhibit randomness greater than that predicted by the Poisson process due to unobserved variables; the weather may have an impact on emergency calls (Chen and Henderson 2001), the state of an organization's IT infrastructure may have an impact on support center calls (Robbins 2007), and TV advertising may have an impact on inbound volume to a sales center (Andrews and Cunningham 1995). Call volume exhibits periodic variability over the course of a day, week, month and year (Andrews and Cunningham 1995; Gans, Koole *et al.* 2003; Robbins 2007). Call center managers attempt to account for these factors when they develop forecasts, yet forecasts may be subject to significant error. Robbins (2007) compares four months of week-day forecasts to actual call volume for 11 call center projects. He finds that the average forecast error exceeds 10% for 8 of 11 projects, and 25% for 4 of 11 projects. The standard deviation of the daily forecast to actual ratio exceeds 10% for all 11 projects. Steckley, Henderson *et al.* (2009) compare forecasted and actual volumes for nine weeks of data taken from four call centers. They show that the forecasting errors are large and modeling arrivals as a Poisson process with the forecasted call volume as the arrival rate can introduce significant error. Robbins, Medeiros *et al.* (2006) use simulation analysis to evaluate the impact of forecast error on performance measures demonstrating the significant impact forecast error can have on system performance.

Some recent papers address staffing requirements when arrival rates are uncertain. Bassamboo, Harrison *et al.* (2005) develop a model that attempts to minimize the cost of staffing plus an imputed cost for customer abandonment for a call center with multiple customer and server types when arrival rates are variable and uncertain. Harrison and Zeevi (2005) use a fluid approximation to solve the sizing problem for call centers with multiple call types, multiple agent types, and uncertain arrivals. Whitt (2006b) allows for arrival rate uncertainty as well as uncertain staffing, *i.e.* absenteeism, when calculating staffing requirements. Steckley, Henderson *et al.* (2004) examine the type of performance measures to use when staffing under arrival rate uncertainty. Robbins and Harrison (2009) develop a scheduling algorithm using a stochastic programming model that is based on uncertain arrival rate forecasts.

The issue of cross training in call centers is summarized in Aksin, Karaesmen *et al.* (2007). The cross training literature for call centers builds on the extensive cross training literature in the context of manufacturing and supply chain operations (Graves and Tomlin 2003; Hopp, Tekin *et al.* 2004; Hopp and Van Oyen 2004). Cross training is relevant in call centers where agents are segregated by skill set and skills-based routing is employed. Issues related to staffing and routing in multi-skill

call centers are summarized in Koole and Pot (2005). Routing issues in the context of call center outsourcing are discussed in Gans and Zhou (2007). Models that address scheduling in multi skill call centers are provided in Avramidis, Chan *et al.* (2007), Avramidis, Gendreau *et al.* (2007), and Cezik and L'Ecuyer (2008). Iravani, Kolfal *et al.* (2007) develop a heuristic to evaluate the effectiveness of different cross training options.

Wallace and Whitt (2005) seek to find the best level of cross training in a call center with multiple call types. In this model there are six call types and every agent is trained to handle a fixed number of those types. The authors use a simulation-based optimization model to find the ideal cross training level. The paper's key insight is that a low level of cross training provides "most" of the benefit. Specifically, they find that training every agent in two skills provides the bulk of the benefit, while additional training has a relatively low payoff. In the Wallace and Whitt (2005) model all agents are cross trained with the same number of skills. Robbins, Medeiros *et al.* (2007) examine the impact of partial pooling in steady state queuing systems with two call types. They find that cross training a small portion of the agents provides most of the benefit. Both of these models ignore the incremental costs associated with cross training and fail to find the optimal cross training level. Chevalier, Shumsky *et al.* (2004) study systems with specialized (single skilled) and fully flexible (multi-skilled) servers, recognizing that fully flexible servers are more costly than specialized servers. They model the call center as a loss system so that customers that cannot be served immediately are lost. They use an approximation procedure to calculate the rate at which calls overflow to flexible servers, and the rate at which overflow calls are lost. Their approximation finds the number of specialized and flexible servers that minimizes the staffing cost given a maximum steady state loss probability, although their approach does not apply an integrality restriction on the number of servers. Based on extensive experimentation they propose an "80/20 rule", whereby 80% of budget dollars are spent on specialized (single skilled) agents and 20% on flexible agents. All these results are consistent with Property 5 in Aksin, Karaesmen *et al.* (2007); "*Well designed limited resource flexibility is almost as good as full resource flexibility in terms of performance*". This suggests that in steady state, cross training more than a moderate proportion of the work force is sub optimal when cross training is costly.

3. Pooling Model

In this section we introduce our model of partial pooling. We first introduce basic terminology and notation used throughout the paper. We assume that in the baseline case the call center is segregated by queue and each queue acts as a separate Erlang-A queuing system. Each queue i receives calls that arrive at a time varying rate $\lambda_i(t)$. The average talk time is $1/\mu_i$.We also assume that callers have exponentially distributed patience with mean $1/\theta_i$, where patience refers to the amount of time a caller is willing to wait on hold. Callers will abandon the queue (hang up) if their call is not answered within their patience time. The call types in each queue are different and require agents with distinct skills.

3.1 Routing

In our model we assume that the call center is staffed by two types of agents. *Base agents* have the skills to service one call type. *Super agents* are cross trained and may serve both call types. We assume that cross-trained agents achieve the same proficiency as single-skilled agents (*e.g.* they have the same service time distribution), although this assumptions is easily relaxed. We assume that the skills based routing system is configured as shown in Figure 3.

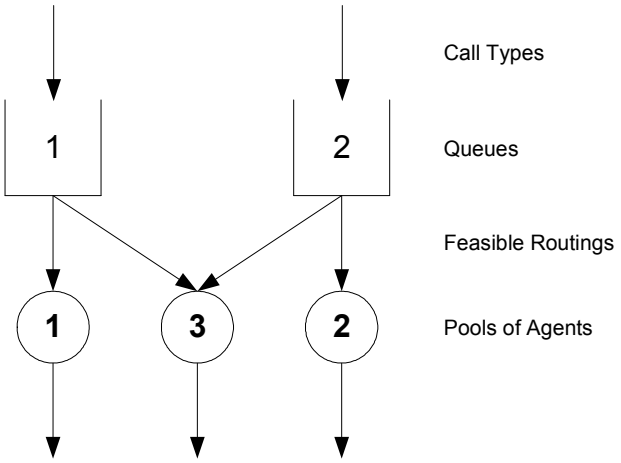

Figure 3 *Basic Routing Structure*

There are two call types, two queues, and three agent pools. Pools 1 and 2 are base agents. Pool 3 is comprised of cross trained agents. Pool 1 agents have Skill 1 and service Queue 1. Similarly Pool 2 agents service Queue 2. Pool 3 agents are cross trained and can service either queue.

We implement a very simple and standard *base agent first* routing model. An incoming call is routed to a base agent if one is available. Only in the case where all base agents are busy is the call routed to a cross-trained super agent. As long as cross-trained agents remain available all calls will be serviced immediately and no abandonment will take place. If no qualified agents are available the call is queued to be served by the next available agent. When base agents become available they take the longest waiting caller from their respective queue. If no calls are waiting they become idle. When cross trained agents become available they take the call from the queue with the most calls. Örmeci (2004) shows that a base agent first routing policy is optimal in a two call type loss system if the specialized servers are at least as fast as flexible servers. Chevalier, Shumsky *et al.* (2004) extend this proof to deal with more than two call types.

3.2 Staffing

Because arrival rates vary considerably call centers must change the staff level throughout the course of the day. During the course of a day agents are assigned to *shifts*, which specify the time they start and end work but do not explicitly account for breaks. A *schedule* specifies the shift and the days of the week an agent works.

The call center has a set of feasible schedules based on staffing policies. A *staff plan* specifies the number of agents assigned to each schedule.

In our call center, project staffing varies from two agents overnight to as many as 70 agents during peak hours. A typical call center may have shifts starting at any time period. However, agents are scheduled to work a specified number of hours, so the call center can not vary the staff as quickly as demand varies. The call center is therefore subject to periods of both tight capacity and excess capacity in any given day.

3.3 System Costs and Objective

Our goal for this call center is to satisfy a service level objective for each call type with a high probability while minimizing overall staffing cost. Because call volume is stochastic it is not practical to meet the service level target with certainty. We therefore chose to implement the service level target as a soft constraint by applying a penalty cost to a realized service level below the target.

We specify an optimization problem with the following definitions:

Sets
I: time periods
J: possible schedules
K: agent types
L: call types

Decision Variables
x_{jk}: number of type k agents assigned to schedule j

Deterministic Parameters
c_{jk}: cost of schedule j for agent type k
a_{ij}: 1 if schedule j is staffed in time i, 0 otherwise
g_l: global TSF SLA goal for call type l
μ_l: minimum number of agents in any period for call type l
r_l: per point penalty cost of TSF shortfall for call type l

Stochastic Inputs
ξ: random arrival and service times

Stochastic Outputs
S_l: realized global TSF for call type l

The optimization problem can then be expressed as

$$\min \sum_{j \in J} \sum_{j \in K} c_{jk} x_{jk} + E_{\xi}\left[\mathrm{Q}(\boldsymbol{x}, \xi)\right] \tag{2.1}$$

Subject to

$$\sum_{j \in J} a_{ij} x_{j1} + \sum_{j \in J} a_{ij} x_{j3} \geq \mu_1 \qquad \forall i \in I \tag{2.2}$$

$$\sum_{j \in J} a_{ij} x_{j2} + \sum_{j \in J} a_{ij} x_{j3} \geq \mu_2 \qquad \forall i \in I \tag{2.3}$$

$$x_{jk} \in \square^{+} \qquad \forall k \in K, j \in J \tag{2.4}$$

The objective function (2.1) is the sum of the deterministic cost of agent staffing plus the expected cost of the implicit penalty cost function $\mathrm{Q}(\boldsymbol{x}, \xi)$. The penalty cost

is a function of the fixed staffing plan $\boldsymbol{x}$, and the random vector ξ of call arrival and service times. The cost function is a non-negative function that calculates a cost associated with failure to achieve the specified SLA. The cost function can be based on any combination of performance metrics desired; such as telephone service factor, average speed to answer, or abandonment. In our analysis we modeled the penalty function $Q(\boldsymbol{x},\xi)$ as a multiple of the shortfall in the TSF SLA, with no benefit realized for over achieving the target service level.

$$Q(\boldsymbol{x},\xi) = r_1(g_1 - S_1)^+ + r_2(g_2 - S_2)^+$$

Note that the objective function does not include a direct cost of abandonment. However, the TSF is defined as the proportion of calls presented that are answered within the time limit. Therefore, the effect of abandonment is to reduce the realized TSF.

Constraint (2.2) specifies the minimum number of agents that must be staffed to handle type 1 calls at any time. The minimum may be a combination of base and cross trained agents. Constraint (2.3) applies the same restriction for type 2 calls.

3.4 Call Arrival Process

Multiple approaches are available for generating simulated arrival patterns. A thorough analysis is provided in Avramidis, Deslauriers *et al.* (2004). For our test problems we use a straight forward two-stage algorithm similar to the model in Weinberg, Brown *et al.* (2007). We use a multiphase, multiplicative model where the arrival rate is the product of a daily number of calls and the proportion of daily calls received in that time period, both of which are random. Details of the algorithm are presented in Figure 4, but it should be noted that the scheduling algorithm is in no way dependent on the model of arrivals.

1. *Generate a call volume for each day of the week using the mean and standard deviation specified for the day.*
2. *For each time period in each day generate a random proportion of call volume based on the specified mean and standard deviation for the time period.*
3. *Normalize the time period proportions so that they add to 1 for each day.*
4. *Calculate the per period call volume by multiplying the daily total by the time period proportion.*

Figure 4 *Simulated Call Generation Algorithm*

4. A Simulation-based Optimization Method

Stochastic models that can be expressed in an analytical format can be solved via a variety of optimization methods. For models where it is difficult to accurately represent the system in closed form, simulation and simulation-based optimization (SBO) can be used. Overviews of SBO are presented in Chapter 12 of Law (2007) and in Fu (2002).

At the most general level, an optimization algorithm has two basic components: generating candidate solutions, and evaluating candidate solutions. In SBO the

solution evaluation step is performed by executing a discrete event simulation (DES) model. Using DES allows us to evaluate a very general model of our stochastic system. The literature on DES is vast; popular texts include Law and Kelton (2000), Banks (2005), and Law (2007). In SBO the bulk of the computational effort is spent on the evaluation step, but from an algorithm design perspective the challenge is developing a method to generate the next candidate solution. A common approach treats the objective function (simulation model) as a black box and simply searches the feasible space for better solutions. These search methods often employ randomization in the search process. There are a wide range of search methodologies available that are classified in the general category of metaheuristics (Fu 2002; Law 2007). Metaheuristics are "solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space" (Glover and Kochenberger (2003). Comprehensive reviews of various metaheuristics are provided in Glover and Kochenberger (2003) and Burke and Kendall (2005). Metaheuristics have been widely applied in deterministic combinatorial optimization problems (Nemhauser and Wolsey 1988; Papadimitriou and Steiglitz 1998; Wolsey 1998). An introductory review of their application to SBO is provided in Fu (2002). Search methodologies include genetic algorithms (Reeves 2003; Sastry, Goldberg *et al.* 2005), Tabu search (Gendreau and Potvin (2005), and simulated annealing (Henderson, Jacobson *et al.* 2003; Aarts, Korst *et al.* 2005). Most metaheuristics implement some form of a neighborhood based search. Given a candidate solution x, the neighborhood $N(x)$ is a set of feasible points that are "close" in some sense to x.

We use a simulation-based local search algorithm guided by a Variable Neighborhood Search (VNS) metaheuristic. VNS is a metaheuristic that makes systematic changes in the neighborhood being searched as the search progresses (Hansen and Mladenovic 2001; Hansen and Mladenovic 2005). When using VNS, a common approach is to define a set of nested neighborhoods, such that

$$N_1(x) \subset N_2(x) \subset ... \subset N_{k_{Max}}(x) \quad \forall x \in X$$

Figure 5 presents a graphical representation.

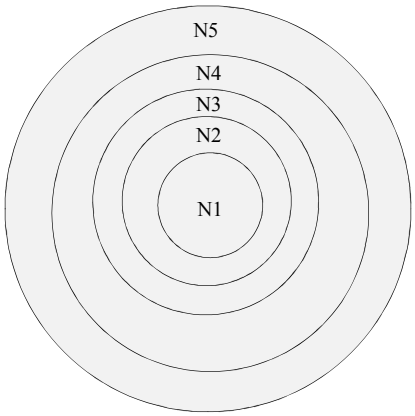

Figure 5 *Nested Neighborhoods*

Neighborhood 1 is the smallest neighborhood, *i.e.* it contains the fewest solutions. Neighborhood 2 is the next largest neighborhood, containing all the solutions in neighborhood 1, plus a set of additional solutions. Neighborhood 5 is the largest neighborhood and each of the other neighborhoods is a subset of neighborhood 5. The search process begins in Neighborhood 1 and only moves outward when an improving solution cannot be found, returning to Neighborhood 1 whenever an improving solution is found. The algorithm terminates when no improving solution can be found in Neighborhood 5.

The general structure of the VNS is then as follows:

1 Initialization

- *Select the set of neighborhood structures* N_k, *for* $k = 1,...,k_{\max}$
- *Construct an initial incumbent solution,* x_I, *using some heuristic procedure.*
- *Select a confidence level* α *for the selection of a new incumbent solution*
- *Set* ***Found*** = *FALSE*

2 Search: repeat the following until Stop=True

- *Find* $n_{k_{\min}}$ *candidate solutions,* x_C *that are neighbors of* x_I
- *Simulate the system with each candidate and compare the results to the incumbent using a pair-wise T Test.*
- *If any* x_C *is superior to* x_I *at the* α *level then set* $x_I = x_C^*$, *where* x_C^* *is the best candidate solution. Set* $k = 1$, *and* ***Found*** = *TRUE*
  Else, set $i = n_{k_{\min}}$, *set* ***Found*** = *FALSE, and repeat until (*$i = n_{k_{\max}}$ *or* ***Found***=*TRUE)*
  - *Find a new candidate* x_{k_i}
  - *Set* $i = i + 1$
  - *Simulate the system with the candidate and compare the results using a pairwise T Test.*
  - *If* x_{k_i} *is superior to* x_I *at the* α *level then set* $x_I = x_{k_i}$, $k = 1$, *and* ***Found*** = *TRUE*
- *If no new incumbent was found in neighborhood* k *then*
  - *Set* $k = k + 1$
  - *If* $k > k_{\max}$ *then* ***Stop*** = *TRUE*

Figure 6 *General VNS Search Algorithm*

This algorithm searches the neighborhood of the current incumbent evaluating at least $n_{k_{\min}}$ points. If no statistically significant improving solution is found, it continues to search the current neighborhood until either an improving solution is found or a total of $n_{k_{\max}}$ points have been evaluated. If no new incumbent is found the search continues in the next largest neighborhood. The search process continues until no improving solution is found in the largest neighborhood structure.

Two important parameters for this search process are $n_{k_{\min}}$ and $n_{k_{\max}}$, the lower and upper bounds on the number of neighbors to evaluate before moving to the next neighborhood. If the neighborhood is defined narrowly, these parameters can both be set equal to the total number of neighbors, and the neighborhood is searched exhaustively. In larger neighborhoods an exhaustive search is not practical and solutions are selected at random. In this case $n_{k_{\min}}$ is the minimum number of neighbors to evaluate.

5. Numerical Experiments on Sample Call Center Projects

In this section we examine the issue of finding optimal staffing plan for pairings of call types with different arrival and talk time characteristics. We apply the VNS approach defined in Section 4 to solve the optimization problem defined in Section 3. The arrival rates and talk time characteristics are based on three sample projects.

5.1 Sample Call Center Projects

Our numerical experiments are based on three sample call types or *projects*; *i.e.,* outsourcing contracts we analyzed. A project represents the calls generated from an outsourcing client. Current operations segregate these projects, allocating them to separate queues each with dedicated staffing and dedicated management. We analyzed multiple projects but selected three for detailed analysis.

Type *J* calls are generated by a corporate help desk for a large industrial company averaging about 750 calls on weekdays. Type *S* calls are from a help desk that provides support to employees of a large national retail chain. Call volume on this desk is about 2,000 calls on weekdays. Because this desk supports users in retail stores, as opposed to corporate offices, the daily pattern of call volumes is quite different from call type *J*. This company was making major changes in its IT infrastructure and as such call volume is very volatile and difficult to forecast. Type *O* calls are from a help desk that provides support to corporate and retail site users at another retail chain. This is a small desk with about 500 calls on weekdays.

Statistical models were developed for each of three sample call types. For each type we eliminated holidays from the data set. The day of week effect was then calculated by estimating the mean and standard deviation of arrivals on each "normal" day. We then estimated the proportion of calls received in each 30-minute period along with the associated standard deviation. A summary of the data for each of these model projects is shown in Table 1.

Table 1 *Call Type Summary*

	Call Type *J*	Call Type *S*	Call Type *O*
Support Base	Corporate	Retail	Corporate/Retail
Hours of Operation	24x7	24x7	24x7
TSF SLA (%/secs)	80/60	80/120	80/120
Average Weekly Volume	3,825	10,600	3,000
Average Talk Time (mins)	12.0	13.5	14.0

While these estimates are based on several months of data, a more accurate model fitting would require a larger data set. Our intent is not to develop specific forecasting models for these projects; rather it is to develop representative and realistic models of projects that can be used to validate the decision models, and to generate insight into the operating characteristics of different classes of projects.

5.2 The Optimization-Simulation Approach

Our goal is to test the three potential pairings of these call types. We seek to find a near optimal staffing plan for the base and cross trained agent pools. To start the process we begin with a staffing plan that includes no cross-training. In this analysis we choose to generate the initial staffing plan for each call type independently using a stochastic optimization program described in Robbins and Harrison (2009). We configured the model to generate a staffing plan at a lower TSF and with a minimum staffing level of one instead of two agents. This procedure creates a staff plan that is slightly understaffed, *i.e.* we staff to a TSF of 75% vs. the required 80% and apply a minimum staffing level of one instead of two agents. The objective is to save time in the search algorithm by creating an initial plan which achieves a basic (but not adequate) level of service.

The next step is to utilize the VNS algorithm described in Section 4 to find a near-optimal staffing plan that includes cross trained agents. Constraints (2.2) and (2.3) are added to the penalty term and the algorithm is configured to flag an error condition if these constraints are not satisfied. We define a nested neighborhood structure with five individual neighborhoods. In our neighborhood structure we make a distinction between active schedules – those schedules with at least one agent assigned in the current incumbent solution, and feasible schedules – all the schedules to which agents can be assigned.

We define the following neighborhoods

- $N_1(x)$: **Active 1 Change**: the set of all staffing plans where an active assignment is incremented by ±1. (*i.e.* randomly select an active schedule x_{jk} and either increment or decrement the staffing level by 1.)
- $N_2(x)$: **Active 2 Change**: the set of all staffing plans where two active assignments are incremented by ±1. (*i.e.* randomly select two active schedules and either increment or decrement the staffing level of each by 1.)
- $N_3(x)$: **Feasible 1 Change**: the set of all staffing plans where a feasible assignment is incremented by ±1. (*i.e.* randomly select a feasible schedule x_{jk} and either increment or decrement the staffing level by 1.)
- $N_4(x)$: **Feasible 2 Change**: the set of all staffing plans where two feasible assignments are incremented by ±1. (*i.e.* randomly select two feasible schedules and either increment or decrement the staffing level of each by 1.)
- $N_5(x)$: **Feasible 3 Change**: the set of all staffing plans where three feasible assignments are incremented by ±1. (*i.e.* randomly select three feasible schedules and either increment or decrement the staffing level of each by 1.)

The neighborhood structure starts small, adding or reducing staffing levels on currently assigned schedules, then expands by considering multiple staffing changes simultaneously, and by considering all feasible schedules. In each neighborhood,

new staffing plans are generated by modifying the current best solution and a large number of alternatives are evaluated in each iteration of the algorithm.

Rather than using a pure random search in each neighborhood, we have implemented heuristic methods to guide the search. In this modified approach each time a new neighbor is required the algorithm generates the neighbor using either the heuristic for that neighborhood or a pure random permutation. Table 2 summarizes the heuristics utilized in each neighborhood and the minimum and maximum number of candidate solutions evaluated.

Table 2 *Neighborhood Search Heuristics*

Neighborhood	Heuristics	$n_{k_{\min}}$	$n_{k_{\max}}$
$N_1(x)$: **Active 1 Change**	• **Pool Support**: select an active schedule in Pool 1 or Pool 2 and staff an agent to the same schedule in the cross trained pool.	20	NA
$N_2(x)$: **Active 2 Change**	• **Cross Train**: select an active schedule in Pool 1 or Pool 2 and change the agent's designation to a cross trained agent. • **Untrain**: select a staffed schedule in Pool 3 and change the designation to either 1 or 2.	5	60
$N_3(x)$: **Feasible 1 Change**	• **Add Max Cover**: find the set of feasible schedules that covers the most short-staffed periods and schedule an agent to one of those schedules.	20	60
$N_4(x)$: **Feasible 2 Change**	• **Active Time Shift**: select an active schedule and shift the assignment forward or backward by one time period. (*i.e.* move the assignment to a schedule that starts 30 minutes earlier or 30 minutes later.)	1	75
$N_5(x)$: **Feasible 3 Change**	• **Two for One**: pick a schedule in Pool 1 or 2, then find the closest active matching schedule in the other pool, decrement each of these assignments and staff a cross trained agent.	1	75

The logic behind this neighborhood structure and its heuristics is relatively straightforward if we recall that the initial incumbent solution is the result of an optimization designed to slightly under staff each base agent pool. First, the set of schedules selected in the optimization process (active schedules) will closely match the time profile of demand. The set of active schedules will typically be a small subset of the feasible schedules. Therefore we chose to search these active schedules first. Since the initial staff plan is understaffed by design, additional staffing, particularly in the super agent pool, will decrease penalty costs more than the associated labor. Neighborhood 1 is small enough that we can search it exhaustively. In Neighborhood 2 we test the benefits of changing an agent's skill designation. By testing both training and untraining, we make sure that the incremental cost of training is justified.

When no improvements can be found in the set of active schedules the search is expanded to the full set of feasible schedules. The heuristic in Neighborhood 3 is designed to address the staffing penalty resulting from not having at least two agents available for each call type in each time period. This heuristic is designed to test all of the schedules with the max cover (*i.e.,* the set of feasible schedules that covers the most short-staffed periods) and will often select a super agent as these agents provide cover for both call types. In Neighborhood 4 we allow for two changes in the feasible schedule and specifically test for the impact of shifting a schedule forward or backward by one time period to potentially better cover a service level gap. The logic of the Neighborhood 5 heuristic is based on the notion that if we have agents in each pool on the same schedule it might be beneficial to replace both of them with a single cross trained agent. This is useful when the service level is being met with high probability, and the penalty is low. Making a two-for-one swap reduces labor cost and may not have a major impact on service level penalties.

In practice the largest number of improving solutions was found in Neighborhood 1. In a typical optimization process improvements are found in three to four neighborhoods, though in some cases all neighborhoods generated improvements. The number of solutions tested at each iteration clearly varies based on where an improvement is found. In our experiment we required that at least 20 candidates were tested before the best was selected. The "max" number varies with the number of active schedules, as Neighborhood 1 is searched exhaustively. In a typical scenario approximately 300 candidate solutions were tested in the final iteration of the algorithm.

The total number of iterations until termination is also random, and depends on the number of feasible schedules. The total number of iterations tended to vary between 15 and 25. Overall this implies that an optimization effort will evaluate somewhere in the range of 500 to 1,500 different schedule combinations.

5.3 Sample Pairing

In this section we analyze the impact of partial pooling under real world situations. We attempt to find optimal plans for cross training agents based on the arrival and talk time characteristics of the three actual outsourcing projects whose summary information is shown in Table 1.

5.3.1 Pooled Optimization – Call Types J and S

First we test the impact of pooling call types *J* and *S*. Call type *J* has relatively stable arrival patterns, while call type S exhibits more volatile arrival patterns. Since call type *S* is retail its busy period extends later into the day than that of call type *J*; call type *S* also has busier weekends and less of a lunchtime lull in call volume than call type *J*.

We implemented our approach by generating preliminary staffing plans for one week with a minimum coverage of one agent and a TSF of 75%. The base agent wage is $10.00 per hour, and $12.50 for cross trained agents reflecting incremental wages and training costs. The cost of failure to meet the TSF is $500 per percentage point below 80% for the week. We implemented the VNS as previously described and simulated each alternative for 10 replications.

Table 3 *Pooled Optimization – Call Types J-S*

	Individual Optimization					Pooled Optimization				Comparison		
Sched Set	Labor Cost	Expected Outcome	TSF 1	TSF2	% Agents Pooled	Labor	Outcome	TSF 1	TSF2	Labor Savings	Total Savings	% Savings
A	41,600	44,504	78.3%	83.5%	13.0%	41,356	42,560	83.2%	83.4%	244	1,944	4.4%
B	40,400	43,529	78.1%	84.7%	15.3%	40,769	41,873	84.4%	83.6%	-369	1,656	3.8%
C	40,320	43,780	78.9%	85.0%	16.1%	40,424	41,171	83.0%	84.0%	-104	2,609	6.0%
D	40,120	43,120	79.4%	84.4%	17.0%	40,732	41,537	83.0%	84.3%	-612	1,583	3.7%
E	40,000	43,240	78.9%	85.3%	18.7%	40,197	41,664	81.4%	83.4%	-197	1,576	3.6%

We utilized common random numbers as a variance reduction technique. The simulation used doubly stochastic Poisson arrivals, exponential service times, and exponential patience times. A confidence level of 80% was used to compare the candidate and incumbent solutions.

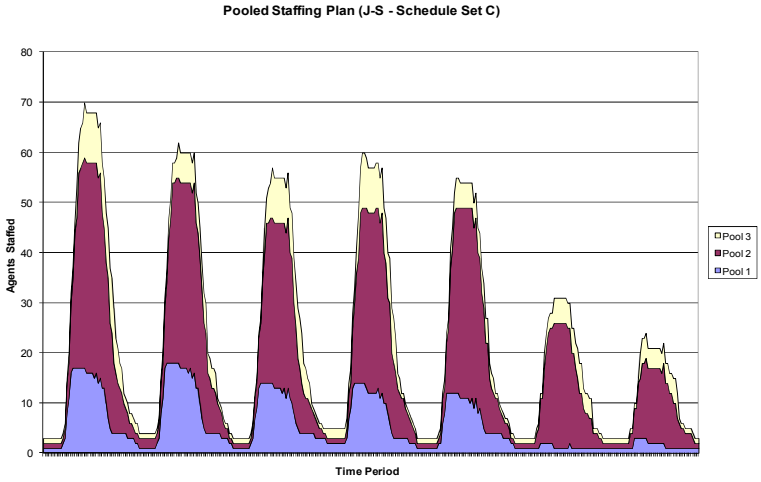

Figure 7 *Pooled Staffing Plan*

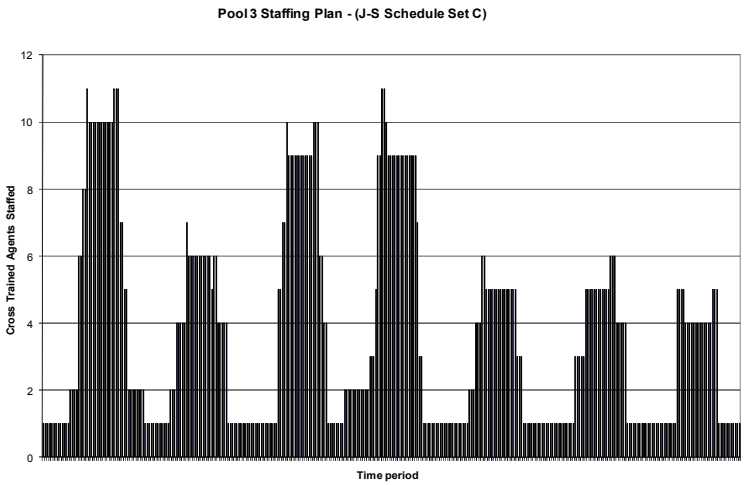

Figure 8 *Cross Trained Agent Staffing Plan*

We compared the results of the pooled optimization to a simulation-based optimization run on each project individually. For each pooling we evaluated five separate levels of shift flexibility detailed in Tables 4 and 5. Table 6 summarizes the results of the pooled optimization effort.

Table 4 *Shift Patterns*

Pattern	Description
5 x 8	5 days a week, 8 hours a day (40 hr week)
4 x 10	4 days a week, 10 hours a day (40 hr week)
4 x 8	4 days a week, 8 hours a day (32 hr week)
5 x 6	5 days a week, 6 hours a day (30 hr week)
5 x 4	5 days a week, 4 hours a day (20 hr week)

Table 5 *Schedule Patterns*

Pattern	Shift Types Included	Feasible Schedules
A	5x8 only	336
B	5x8, 4x10	1,680
C	5x8, 4x10, 4x8	3,024
D	5x8, 4x10, 4x8, 5x6	3,360
E	5x8, 4x10, 4x8, 5x6, 5x4	3,696

The results show that even with a 25% premium for pooled agents, pooling reduces the overall cost of operation. Cost savings vary from 4.4% to 6.0% depending on the scheduling set option. In each case the number of labor hours drawn from the cross trained pool is less than 20%. Note that call type *J*, the smaller volume type, sees an improvement in average service level in each case while the service level for call type *S* remains constant or declines slightly. Intuitively, without pooled agents call type *S* must carry safety capacity to hedge against costly spikes, which is evident by the average service level cushion of 3%-5% for the individual project. In the pooled case spare capacity can be allocated to call type *J* as necessary and each call type has an average service level just above the targeted level. Further insight can be gleaned from the graphical views of the resulting staff plan. In the figures 7 and 8 we plot the staffing plan for schedule set *C*.

Cross trained agents are scheduled throughout the week but are most heavily deployed during the busy periods.

5.3.2 Pooled Optimization Call Types *J-O*

Similar results are found for the pairing of call types *J* and *O* as shown in Table 6.

Table 6 *Pooled Optimization – Call Types J-O*

	Individual Optimization					Pooled Optimization					Comparison	
Sched Set	Labor Cost	Expected Outcome	TSF 1	TSF2	% Agents Pooled	Labor	Outcome	TSF 1	TSF2	Labor Savings	Total Savings	% Savings
A	23,200	24,606	78.3%	79.9%	14.3%	23,228	23,938	80.8%	81.2%	-28	668	2.7%
B	22,800	24,643	78.1%	78.5%	14.5%	22,834	23,547	81.7%	81.4%	-34	1,096	4.4%
C	22,800	24,597	78.9%	78.3%	21.2%	23,115	23,504	81.8%	82.3%	-315	1,093	4.4%
D	22,540	24,396	79.4%	79.7%	19.0%	23,143	23,758	80.7%	82.8%	-603	638	2.6%
E	22,460	24,513	78.9%	79.1%	18.8%	22,698	23,550	80.8%	81.5%	-238	963	3.9%

In this case the savings are slightly less, in the range of 2.7% - 4.4% and the proportion of agents cross trained is slightly higher. In each case labor costs are increased slightly resulting in a higher level of confidence that the service level goal will be achieved. Recalling that these call types have approximately equal volume, the benefits are roughly equally distributed. The average service level for each call type moves up from just below the target to just above the target. Intuitively, since the incremental capacity can be allocated to either call type as needed, the cost of incremental labor is offset by the reduction in penalty costs.

5.3.3 Pooled Optimization Call Types *S-O*

In this final pairing we examine a pooling of call type *S* and call type *O*, both of which have retail oriented periodic patterns. The results are summarized in Table 7.

As in the previous case pooling reduces cost of operation for these call types around 5% by pooling 10%-15% of agents. But unlike the two previous cases, this situation reduces total cost by reducing labor. The intuition is that each of these call types are relatively volatile and must carry significant spare capacity to hedge against uncertainty. By pooling, call type spare capacity can be shared and the total amount of spare capacity is reduced.

Table 7 *Pooled Optimization – Call Types S-O*

	Individual Optimization					Pooled Optimization				Comparison		
Sched Set	Labor Cost	Expected Outcome	TSF 1	TSF2	% Agents Pooled	Labor	Outcome	TSF 1	TSF2	Labor Savings	Total Savings	% Savings
A	41,600	44,387	83.5%	79.9%	10.1%	40,654	42,349	82.4%	80.4%	946	2,038	4.6%
B	40,800	44,424	84.7%	78.5%	13.7%	39,370	41,523	81.2%	80.6%	1,430	2,901	6.5%
C	40,400	44,378	85.0%	78.3%	15.4%	40,034	41,966	82.8%	80.3%	366	2,412	5.4%
D	40,540	44,177	84.4%	79.7%	14.5%	39,768	42,103	82.8%	79.8%	772	2,074	4.7%
E	40,620	44,294	85.3%	79.1%	13.7%	40,273	42,188	82.5%	80.7%	347	2,106	4.8%

5.4 The Impact of Cross Training Wage Differential

The analysis shows that cross training a portion of the workforce can reduce costs even if cross training agents is expensive. In the analysis so far we have assumed that cross training creates a 25% cost premium. In this section we examine the impact of varying the differential for wages and training. We maintain the base agent wage at $10.00 per hour, but we test cross trained agent rates of $11.25, $12.50, and $13.75.

The results are summarized in Table 8. Overall we find that cross training is a viable tactic over this range of costs. The expected savings is naturally declining in the wage and training differential as is the proportion of agents cross trained – although the proportion of agents cross trained is less sensitive to the differential than one might expect.

5.5 Conclusions

Evaluation of these three call type pairings shows that the ability to reduce operating costs by partial pooling is robust across different combinations. The overall savings is approximately 5% with a pooling of approximately 15% of agents. These results are consistent across pairings. However, the mechanism by which the savings are obtained is different. In some cases the aggregate service level is increased when adding more (pooled) agents allows efficient improvement in service level goal attainment. In other cases pooling allows redundant capacity to be reduced through efficient sharing of spare capacity.

Table 8 *The Impact of Wage Premiums on Cross Training Results*

		No Cross Training	Cross Training Wage Differential					
			$11.25		$12.50		$13.75	
Pairing	Sched Set	Expected Outcome	% Agents Pooled	% Savings	% Agents Pooled	% Savings	% Agents Pooled	% Savings
J-S	A	44,504	15.3%	7.1%	13.0%	4.4%	14.3%	3.9%
	B	43,529	17.3%	5.7%	15.3%	3.8%	13.3%	3.7%
	C	43,780	15.9%	6.9%	16.1%	6.0%	15.1%	4.0%
	D	43,120	19.0%	5.4%	17.0%	3.7%	16.4%	2.6%
	E	43,240	19.4%	5.5%	18.7%	3.6%	17.4%	0.9%
J-O	A	24,606	14.3%	4.1%	14.3%	2.7%	10.7%	0.9%
	B	24,643	19.6%	5.5%	14.5%	4.4%	16.1%	1.5%
	C	24,597	22.9%	5.8%	21.2%	4.4%	15.4%	2.5%
	D	24,396	28.3%	5.4%	19.0%	2.6%	14.9%	0.9%
	E	24,513	20.1%	6.3%	18.8%	3.9%	18.3%	0.6%
S-O	A	44,387	9.1%	6.3%	10.1%	4.6%	6.1%	5.2%
	B	44,424	18.2%	5.9%	13.7%	6.5%	14.4%	3.3%
	C	44,378	15.9%	7.4%	15.4%	5.4%	13.9%	3.4%
	D	44,177	16.5%	6.1%	14.5%	4.7%	13.0%	3.3%
	E	44,294	17.5%	5.6%	13.7%	4.8%	16.7%	1.9%

6. Summary and Future Research

In this paper we examine the concept of partial pooling of agents in call centers. The basic premise is that in cases where training is expensive, it is not practical to train all agents to handle multiple call types. We investigate the option of training some agents to handle two call types and show that this approach can yield substantial benefits.

This paper makes a contribution by evaluating a pooling approach not previously analyzed. Wallace and Whitt (2005) find that training every agent in two skills provides the bulk of the benefit, while additional training has a relatively low payoff. Although the general finding in our paper is similar, *e.g.* small levels of cross training give the majority of the benefit, our model assumes that only a small proportion of agents are cross trained. In our model we include the cost of cross training and seek an optimal level. Our analysis focuses on the case where both arrival rates and staff levels change dramatically during the course of the SLA period. We are very interested in how the variable fit of capacity to load impacts the benefit of partial pooling.

The clear implication for managers is that cross training a limited number of agents is a cost effective option under a wide range of assumptions and conditions. The model presented here provides a specific method for finding the appropriate level of cross training, but also provides some basic insight. Managers should seek to cross train a moderate level of the agent base to support multiple call streams. In the case of multilingual call centers, managers need a few multilingual agents, but don't need all agents to be multilingual.

Several extensions to this model are possible. While our model assumed base agents and cross trained agents are equally productive, we might want to consider the

possibility that different agent types might have different levels of productivity. In addition, we may wish to consider larger skill poolings. Multi-lingual call centers, for example, will often support a large number of languages with agents possessing a mix of multi-lingual skills.

7. References

1. Aarts, E., J. Korst and W. Michiels (2005). Simulated Annealing. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. E. K. Burke and G. Kendall. New York, NY, Springer**:** 187-210.
2. Aksin, Z., M. Armony and V. Mehrotra (2007). "The Modern Call-Center: A Multi-Disciplinary Perspective on Operations Management Research." *Production and Operations Management* **16**(6): 665-668.
3. Aksin, Z., F. Karaesmen and E. L. Ormeci (2007). A Review of Workforce Cross-Training in call centers from an operations management perspective. *Workforce Cross Training Handbook*. D. Nembhard, CRC Press.
4. Andrews, B. H. and S. M. Cunningham (1995). "L.L. Bean Improves Call-Center Forecasting." *Interfaces* **25**(6): 1.
5. Atlason, J., M. A. Epelman and S. G. Henderson (2004). "Call center staffing with simulation and cutting plane methods." *Annals of Operations Research*: 333-358.
6. Atlason, J., M. A. Epelman and S. G. Henderson (2008). "Optimizing Call Center Staffing Using Simulation and Analytic Center Cutting-Plane Methods." *Management Science* **54**(2): 295-309.
7. Avramidis, A. N., W. Chan and P. L'Ecuyer (2007). Staffing multi-skill call centers via search methods and a performance approximation, University of Montreal.
8. Avramidis, A. N., A. Deslauriers and P. L'Ecuyer (2004). "Modeling Daily Arrivals to a Telephone Call Center." *Management Science* **50**(7): 896-908.
9. Avramidis, A. N., M. Gendreau, P. L'Ecuyer and O. Pisacane (2007). *Simulation-Based Optimization of Agent Scheduling in Multiskill Call Centers*. 2007 Industrial Simulation Conference.
10. Banks, J. (2005). *Discrete-event system simulation*. Upper Saddle River, N.J., Pearson Prentice Hall.
11. Baron, O. and J. M. Milner (2006). Staffing to Maximize Profit for Call Centers with Alternate Service Level Agreements**:** 33.
12. Bassamboo, A., J. M. Harrison and A. Zeevi (2005). "Design and Control of a Large Call Center: Asymptotic Analysis of an LP-based Method." *Operations Research* **54**(3): 419-435.
13. Brown, L., N. Gans, A. Mandelbaum, A. Sakov, S. Haipeng, S. Zeltyn and L. Zhao (2005). "Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective." *Journal of the American Statistical Association* **100**(469): 36-50.
14. Burke, E. K. and G. Kendall, Eds. (2005). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. New York, NY, Springer.

15. Cezik, M. and P. L'Ecuyer (2008). "Staffing Multiskill Call Centers via Linear Programming and Simulation." *Management Science* **54**(2): 310-323.
16. Chen, B. P. K. and S. G. Henderson (2001). "Two Issues in Setting Call Centre Staffing Levels." *Annals of Operations Research*108 (1): 175-192.
17. Chevalier, P., R. A. Shumsky and N. Tabordon (2004). Routing and Staffing in Large Call Centers with Specialized and Fully Flexible Servers**:** 38.
18. Fu, M. C. (2002). "Optimization for Simulation: Theory vs. Practice." *INFORMS Journal on Computing* **14**(3): 192-215.
19. Fukunaga, A., E. Hamilton, J. Fama, D. Andre, O. Matan and I. Nourbakhsh (2002). *Staff Scheduling for Inbound Call Centers and Customer Contact Centers*. Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada.
20. Gans, N., G. Koole and A. Mandelbaum (2003). "Telephone call centers: Tutorial, review, and research prospects." *Manufacturing & Service Operations Management* **5**(2): 79-141.
21. Gans, N. and Y.-P. Zhou (2007). "Call-Routing Schemes for Call-Center Outsourcing." *Manufacturing & Service Operations Management* **9**(1): 33-51.
22. Garnett, O., A. Mandelbaum and M. I. Reiman (2002). "Designing a Call Center with impatient customers." *Manufacturing & Service Operations Management* **4**(3): 208-227.
23. Gendreau, M. and J.-Y. Potvin (2005). Tabu Search. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. E. K. Burke and G. Kendall. New York, NY, Springer**:** 165-186.
24. Glover, F. and G. A. Kochenberger (2003). *Handbook of metaheuristics*. Boston, Kluwer Academic Publishers.
25. Graves, S. C. and B. T. Tomlin (2003). "Process Flexibility in Supply Chains." *Management Science* **49**(7): 907-919.
26. Hansen, P. and N. Mladenovic (2001). "Variable neighborhood search: Principles and applications." *European Journal of Operational Research* **130**(3): 449-467.
27. Hansen, P. and N. Mladenovic (2005). Variable Neighborhood Search. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. E. K. Burke and G. Kendall. New York, NY, Springer**:** 211-238.
28. Harrison, J. M. and A. Zeevi (2005). "A Method for Staffing Large Call Centers Based on Stochastic Fluid Models." *Manufacturing & Service Operations Management* **7**(1): 20-36.
29. Henderson, D., S. H. Jacobson and A. W. Johnson (2003). The Theory and Practice of Simulated Annealing. *Handbook of metaheuristics*. F. Glover and G. A. Kochenberger. Boston, Kluwer Academic Publishers.
30. Hopp, W. J., E. Tekin and M. P. Van Oyen (2004). "Benefits of Skill Chaining in Serial Production Lines with Cross-Trained Workers." *Management Science* **50**(1): 83-98.
31. Hopp, W. J. and M. P. Van Oyen (2004). "Agile Workforce Evaluation: A Framework for Cross-training and Coordination." *IIE Transactions* **36**(10): 83-98.

32. Iravani, S. M. R., B. Kolfal and M. P. Van Oyen (2007). "Call-Center Labor Cross-Training: It's a Small World After All." *Management Science* **53**(7): 1102-1112.
33. Koole, G. and A. Pot (2005). An Overview of Routing and Staffing in Multi-Skill Contact Centers, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands**:** 1-32.
34. Koole, G. and E. van der Sluis (2003). "Optimal shift scheduling with a global service level constraint." *IIE Transactions* **35**: 1049-1055.
35. Law, A. M. (2007). *Simulation modeling and analysis*. Boston, McGraw-Hill.
36. Law, A. M. and W. D. Kelton (2000). *Simulation modeling and analysis*. Boston, McGraw-Hill.
37. Mandelbaum, A. and S. Zeltyn (2004). Service Engineering in Action: The Palm/Erlang-A Queue, with Applications to Call Centers Draft, December 2004.
38. Milner, J. M. and T. L. Olsen (2008). "Service-Level Agreements in Call Centers: Perils and Prescriptions." *Management Science* **54**(2): 238-252.
39. Nemhauser, G. L. and L. A. Wolsey (1988). *Integer and combinatorial optimization*. New York, Wiley.
40. Örmeci, E. L. (2004). "Dynamic Admission Control in a Call Center With One Shared and Two Dedicated Service Facilities." *IEEE Transactions on Automatic Control* **49**(7): 1157-1161.
41. Papadimitriou, C. H. and K. Steiglitz (1998). *Combinatorial optimization: algorithms and complexity*. Mineola, N.Y., Dover Publications.
42. Reeves, C. (2003). Genetic Algorithms. *Handbook of metaheuristics*. F. Glover and G. A. Kochenberger. Boston, Kluwer Academic Publishers.
43. Robbins, T. R. (2007). Managing Service Capacity Under Uncertainty - Unpublished PhD Dissertation (http://personal.ecu.edu/robbinst/), Pennsylvania State University**:** 240p.
44. Robbins, T. R. and T. P. Harrison (2009). Call Center Scheduling with Uncertain Arrivals and Global Service Level Agreements, East Carolina University**:** 35p.
45. Robbins, T. R., D. J. Medeiros and P. Dum (2006). *Evaluating Arrival Rate Uncertainty in Call Centers*. Proceedings of the 2006 Winter Simulation Conference, Monterey, CA.
46. Robbins, T. R., D. J. Medeiros and T. P. Harrison (2007). *Partial Cross Training in Call Centers with Uncertain Arrivals and Global Service Level Agreements*. Proceedings of the 2007 Winter Simulation Conference, Washington, DC.
47. Sastry, K., D. Goldberg and G. Kendall (2005). Genetic Algorithms. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. E. K. Burke and G. Kendall. New York, NY**:** 97-126.
48. Steckley, S. G., S. G. Henderson and V. Mehrotra (2009). "Forecast Errors in Service Systems." *Probability in the Engineering and Informational Sciences*(23): 305-332.
49. Steckley, S. G., W. B. Henderson and V. Mehrotra (2004). Service System Planning in the Presence of a Random Arrival Rate, Cornell University.

50. Wallace, R. B. and W. Whitt (2005). "A Staffing Algorithm for Call Centers with Skill-Based Routing." *Manufacturing & Service Operations Management* **7**(4): 276-294.
51. Weinberg, J., L. Brown and J. R. Stroud (2007). "Bayesian Forecasting of an Inhomogeneous Poisson Process with Applications to Call Center Data." *Journal of the American Statistical Association* **102**(480): 1185-1198.
52. Whitt, W. (2006a). "Sensitivity of Performance in the Erlang A Model to Changes in the Model Parameters." *Operations Research* **54**(2): 247-260.
53. Whitt, W. (2006b). "Staffing a Call Center with Uncertain Arrival Rate and Absenteeism." *Production and Operations Management* **15**(1): 88-102.
54. Wolsey, L. A. (1998). *Integer programming*. New York, J. Wiley.