# ANALYZING SKILL-BASED ROUTING CALL CENTERS USING DISCRETE-EVENT SIMULATION AND DESIGN EXPERIMENT

Thomas A. Mazzuchi

Department of Engineering Management
and Systems Engineering
The George Washington University
1776 G Street NW
Washington, D.C. 20052, U.S.A.

Rodney B. Wallace

IBM Global Services
7907 Woodbury Drive
Silver Spring, MD 20910, U.S.A.

## ABSTRACT

Call center customer service representatives (CSRs) or agents tend to have different skills. Some CSRs can handle one type of call, while other CSRs can handle other types of calls. Advances in automatic call distributors (ACDs) have made it possible to have skill-based routing (SBR) which is the protocol for online routing of incoming calls to the appropriate CSRs. At present, very little is known about SBR. We develop a discrete-event simulation model to analyze the performance of a $M_n/M_n/C/K$ SBR environment in which incoming calls are handled in priority order and in a non-preemptive manner. We use the design of experiment framework to conduct our analysis. We show empirically that the scenario in which agents have 2 skills is almost as efficient as the scenario where agents have all skills (resource pooling). Also, we discover that no interaction exists between call rate factors when resource pooling exists.

## 1 INTRODUCTION

Call center customer service representatives (CSRs) or agents tend to have different skills. Some CSRs can handle one type of call, while other CSRs can handle other types of calls. Advances in automatic call distributors (ACDs) have made it possible to have skill-based routing (SBR) which is the protocol for online routing of incoming calls to the appropriate agent. At present, very little is known about SBR. According to Gans, Koole, and Mandelbaum (2003), "... the technology has raced ahead of managers' and academics' understanding of how it may best be used, and the characterization of effective strategies for skill-based routing is an open question at all levels of the capacity-planning hierarchy ..." This paper provides insights into the performance of call centers with skill-based routing in which calls are handled in a non-preemptive (NPRP) priority manner.

Skill-based routing is an active research topic, but so far there is only a relative small body of literature as cited in Wallace and Whitt (2004). We have two main objectives in this paper: (1) conduct a general performance assessment of a $M_n/M_n/C/K/NPRP$ SBR call center environment and (2) confirm the resource pooling phenomenon observed in Wallace and Whitt (2004). *Full* or *complete resource pooling* occurs when all agents have all skills. In Wallace and Whitt (2004), it is shown using one-factor-at-a-time SBR analysis that the system where agents have two skills performs nearly as well as the system where agents have all skills (resource pooling).

We will accomplish our objectives by using discrete-event simulation and the design of experiment (DOE) framework to conduct our performance assessment. By conducting factorial experiments, we are able to observe the performance behavior of the system, as well as, identify several different ways of characterizing the existence of resource pooling. For example, we show that no interactions between different call rate factors exist while the system is experiencing resource pooling. *Interaction* between factors occur when we find that the difference in response between the levels of one factor is not the same at all levels of the other factors.

## 2 THE CALL CENTER MODEL

We consider a call center or multi-server queueing system with $C$ total agents, $n$ different call types and telephone trunkline capacity $C+K$. Calls have types $1, \ldots, n$ and are handled in a non-preemptive priority (NPRP) order. The priorities are associated with the skill levels of the agents. Agents have skills at various skill levels (primary, secondary, tertiary and so on). The number of different skill levels is equal to the number of call types $n$. Skill level 1 represents a primary skill. Skill level 2 represents a secondary skill level, and so on. Each agent has one and only one primary skill. Agents may have up to $n-1$ secondary skills at unique skill levels.

Agents with the same primary skill $j$ make up the work group $j$. The number of agents in work group $j$ is denoted by $C_j$, where $1 \leq j \leq m$. The agent skills are given and are represented by a $C \times n$ agent-skill matrix $A$. Each entry of the agent-skill matrix $A$ is

$$a_{ij} = \begin{cases} q & \text{when agent } i \text{ supports call type } q \\ & \text{ at skill level } j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $i = 1, \ldots, C$, $1 \leq q \leq n$, and $1 \leq j \leq n$. Thus, the rows of the agent-skill matrix represent the unique agent identification (ID), the columns represent the skill level (column 1 indicates primary skill, column 2 secondary, and so on), and the entry $a_{ij}$ indicates the call type supported. The agent skill matrix is one of the major components that distinguishes our SBR call center.

In the $C+K$ trunklines capacity, the parameter $K$ is the number of waiting spaces or buffers to hold waiting callers. If an arriving caller finds that there are already $C+K$ customers present, then the caller is blocked and is lost to the system. Callers arriving to each work group are handled or processed using a *first-come first-serve* (FCFS) service discipline among qualified agents. Customer abandonments, retrials, and jockeying are not permitted.

There are two fundamental problems we must address in our SBR call center model construction. They are:

1. What to do When an Arrival Occurs
2. What to do When an Agent Becomes Free.

We address each of theses in the following subsections.

## 2.1 What to do When an Arrival Occurs

To address the issue of what to do when an arrival occurs, we consider the call routing strategy most commonly used by call center managers, the *longest idle agent routing* policy. The longest-idle-agent routing (LIAR) policy sends calls to the agents that have been waiting the longest for a call since the completion of their last job (i.e., idle the longest). This policy is considered to be a *fair* scheme since it is well-known to balance the call volume across all agents. To adjust for priorities, the LIAR policy that we adopt sends calls to the agents that have been waiting the longest (or idle the longest) and have the highest skill-level to handle the call.

## 2.2 What to do When an Agent Becomes Free

When agents become free, if there are no customers in the $n$ queues then the agents go idle; otherwise, the first customer in the queue that the agents can support at their highest skill level is taken into service. More precisely, when agents become free, the first customer in the queue in which the agents have a primary skill level to support the call is taken into service. We will refer to this queue as the agents primary skill queue. If there are no customers in the agents primary skill queue, the first customer in the agents secondary skill queue is taken (provided the agents have a secondary skill or $a_{i2} \neq 0$). The process is continued in this manner until either a customer is found that the agents can support or all skill levels have been exhausted. If the agents cannot find a waiting customer that they can support then the agents go idle. Customers waiting in the queue that are not supported by freed or idled agents continue to wait until agents that can support their call type become available.

## 2.3 Probability Assumptions

This section captures the stochastic assumptions of our $M_n/M_n/C/K/NPRP$ call center which involve the arrival process and service time distribution.

Callers or customers arrive to the call center in accordance to a Poisson with rate $\lambda$. These callers then, independently of one another, select the type of service desired with probability $p_i$ where $i$ indicates the type of service requested or simply the customer's type, $i = 1, \ldots, n$. Thus, the arrival process for each call type is characterized by a Poisson process with rate $\lambda_i \equiv \lambda p_i$ for $i = 1, \ldots, n$ where $\lambda = \sum_{j=1}^{n} \lambda_j$.

The service times to process calls depend only on the call type and are independent and identically distributed (IID) with exponential service times $1/\mu_i$ where $i$ represents the call type for $i = 1, \ldots, n$. Thus, we are assuming that the difference in call handling capability of newly trained agents and experienced agents can be ignored. In other words, we assume that if agents have the appropriate skill to handle the customer call as predefined in the agent-skill matrix then the agents can process the call within essentially the same amount of time.

In our model construction, we assume that time agents spend handling after-call work is included in the mean service time. This simplifying assumption ignores the fact that the trunklines become available while the agents are busy with after-call work. Harris, Hoffman, and Saunders (1987) have shown that the impact of combining talk-time and after-call work time is not significant enough to warrant separate models for the two components.

## 2.4 Two Problems

Given the $M_n/M_n/C/K/NPRP$ skill-based routing call center model described in this chapter, we will focus our attention on two specific and fundamental areas:

1. Performance Analysis
2. Resource Pooling.

We will discuss each of these in turn.

### 2.4.1 Performance Analysis

First, we would like to understand the performance of skill-based routing with non-preemptive priorities in general. The performance analysis problem can be stated simply as follows. Given the offered load $\rho = \lambda/\mu$, trunk line capacity $C + K$, the routing policy, service level target time $\tau$, and the agent skill profile, what are the various performance metrics of interest. In particular, we would like to understand the eight (8) key call center performance measures listed in Table 1.

In the table, we have several random variables of interest. They are the number of callers in the system, denoted by $Q$, the aggregate delay experienced by callers, denoted by $D$, and the delay experienced by callers requesting type $i$ service, denoted by $D_i$. The number of callers in the system includes the number of callers in service plus the number of callers in queue or waiting. In the table, the indices $i$ and $j$ indicate the call type or service request ($i = 1, \ldots, n$) and the work group ID ($j = 1, \ldots, m$).

Table 1: SBR Call Center Performance Measures of Interest

| Performance Measure | Description |
|---|---|
| 1. $\mathsf{P}(Q = C + K) = \epsilon$ | Probability of blocking |
| 2. $\mathsf{E}[D \mid Q < C + K] = W$ | Average speed to answer given system entry |
| 3. $\mathsf{E}\left[D_i \mid Q < C + K\right] = W_i$ | Average speed to answer call type $i$ given system entry |
| 4. $\mathsf{P}(D \le \tau \mid Q < C + K) = 1 - \delta$ | Percent of calls that are answered within $\tau$ minutes given system entry |
| 5. $\mathsf{P}(D_i \le \tau \mid Q < C + K) = 1 - \delta_i$ | Percent of calls of type $i$ that are answered within $\tau$ minutes given system entry |
| 6. $\upsilon$ | Agent utilization |
| 7. $\upsilon_j$ | $j$th work group utilization |
| 8. $\upsilon_j^*$ | $j$th work group primary skill utilization |

In Table 1, the first performance metric, the probability that an arriving caller is blocked, is a measure of the call center's availability and is sometimes apart of the service level agreements (SLAs). The second parameter $\mathsf{E}[D \mid Q < C + K]$ and the fourth $\mathsf{P}(D \le \tau \mid Q < C + K)$ are speed-to-answer performance measures and are typically apart of the service levels as well. These two aggregate quantities are conditioned given admission or entry into the system. Usually, one of the two and not both speed-to-answer metrics is apart of the SLA. Average speed to answer (ASA) is the call center term reserved for $\mathsf{E}[D \mid Q < C + K]$. Both Speed-to-answer and availability SLAs drive staffing and equipment (trunklines) requirements. Massey and Wallace (2004) developed asymptotic-based algorithms to determine optimal $(C, K)$ in a $M/M/C/K$ queue while holding the SLAs for blocking and the conditional probability of delay fixed.

The last three measures of performance deal specifically with tracking agent's utilization. The *average utilization* for an agent is the percent of time that he/she is busy processing calls or one minus the fraction of time he/she is idle.

### 2.4.2 Resource Pooling

A second area of focus for us is to understand the phenomenon of *resource pooling*. In our skill-based routing call center environment, agents are flexible and can support multiple skills. If agents have only one skill in an SBR environment in which there are $n$ different work groups then it is well-known that system will behave as a collection of much smaller independent call centers (assuming blocking is negligible). At the other extreme, if each agent can support all service requests or skills (call center term reserved for *universal agent*), then the system behaves as one big call center or single multi-server system. Under this big call center scenario, there is no situation in which there are waiting customers and idle agents. When this situation occurs, we say that the system exhibits full resource pooling or simply resource pooling. We are seeking to understand the minimum number of skills agents needed for the system to behave as if full resource pooling exists.

By conducting resource-pooling experiments, Wallace and Whitt (2004) showed that indeed resource pooling exist when agents have two (2) skills each. The resource-pooling experiments involved one-factor-at-a-time analysis. In this paper, our goal is to show how factorial design experiments can uncover this fact, as well as describe the level of interaction between key variables or factors.

## 3 THE SIMULATION MODEL

Using the C programming language, we develop a discrete-event simulation to analyze the $M_n/M_n/C/K/NPRP$ skill-based routing call center model described in the previous section. We consider many of the suggested queueing simulation techniques of Law and Kelton (2001), Ross (2002), Gross and Harris (1998), and Whitt (1989) in constructing a very robust multi-server simulation model. In addition, we have also developed many techniques for dealing with the unique aspects of skill-based routing.

From stochastic output processes of the form $\{X_n : n \ge 1\}$, we estimate the steady-state performance measures shown in Table 1 using one long simulation run. During the long run, we delete an initial portion of the observations in order to account for the effects due to initial bias. We choose the initial portion to delete large enough so the system is nearly in steady-state. The remaining observations are divided into a fixed number of non-overlapping batches of equal length. The length of the batch size is sufficiently large such that correlation between batches become negligible and the batch mean approximately follows a normal distribution. We use sample batch means to estimate variances and construct confidence intervals.

We investigate a number of techniques to validate and verify our SBR simulation program and its output. First, we validated our SBR call center model assumptions and logic with call center managers and experts. We acknowledge that this level of validation is limited and that the ultimate goal of the validation process is to compare our simulation output against outputs from real-world systems. In verifying our simulation, we followed a number of industry-accepted verification techniques: modular testing, sensitivity testing, stress testing, trace analysis, and output comparison against known models. See Wallace (2004) for details.

Table 2 shows how our SBR simulation output (95% confidence interval) compares against five different and known queueing models: (1) the $M/M/C/0$ queue, (2) two separate $M/M/C/\infty$ models, (3) the $M/M/C/K$ queue, (4) Ridley's discrete-event simulation model ($M_2(t)/M_2/C/\infty/NPRP$) taken from Ridley (2004), and (5) Stanford and Grassmann (1998) bilingual call center model ($M_2/M/20/\infty/NPRP$ SBR). In every scenario, we generate approximately 1,000,000 observations in which the first 200,000 observations are discarded, the initial deletion period $l$. To achieve higher precision (i.e., smaller confidence intervals), we need only to increase the number of overall observations. However, we should note that all of the key performance measures are within 3% of our estimators and fall within our 95% confidence intervals. These results provide us with a high level of confident that our discrete-event simulation outputs are consistent with what we anticipate in a call center environment with skill-based routing.

## 4 MIXED-FACTORIAL DESIGN EXPERIMENT

In this section, we formulate our skill-based routing call-center problem as a mixed-factorial design experiment using the suggested techniques of Jain (1991) and Montgomery (2001). Figure 1 shows the general framework of design of experiment - DOE (e.g., controllable and uncontrollable factors, response variables). It is important to note that general queueing models are not necessarily *black boxes*, representing systems for which we have no ideas about how factors affect the response variables as illustrated in the top drawing in Figure 1. In fact, we have a wealth of knowledge about the way the factors listed in middle drawing of Figure 1 affect performance measures in general. We elect to use the DOE framework to thoroughly describe specific performance behavior of the SBR call center model under study. In addition, we will investigate the factor interactions.

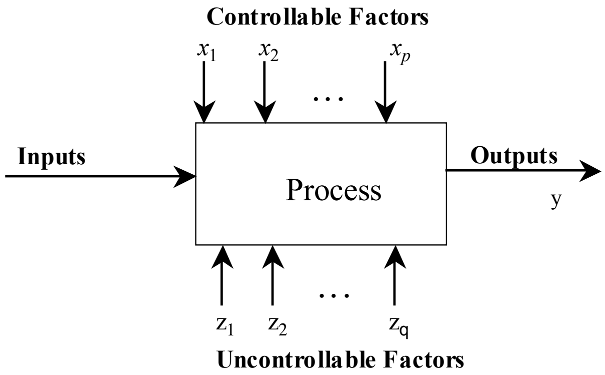

Figure 1: The General DOE Framework

In our call center, the input factors are the rates in which the different call requests enter the system, $\lambda_i$ ($i = 1, \ldots, n$). In general, call volumes for inbound call centers are are uncontrollable (although they may be controllable for the purposes of a test). The output or response variables are the $2(n+m)+4$ performance measures shown in Table 1, where $n$ is the number of call types and $m$ is the number of work groups. Other uncontrollable factors are the times agents spend with callers or mean service times, $1/\mu_i$ ($i = 1, \ldots, n$).

Three controllable factors are the routing policy used to forward calls to agents, the number of trunk lines $C+K$ and the number of skills agents have as defined in the agent skill matrix $A_{C\times n}$. The agent skill matrix also contains very key quantitative information such as the total number of agents $C$ and the number of agents in each work group $C_j$ ($j = 1, \ldots, m$). Each entry of the agent-skill matrix $A$ is captured in formula (1).

In our DOE, we fix the following factors: the number of total agents $C$, the trunkline capacity $C+K$, the mean service times (i.e., $1/\mu_1 = \ldots = 1/\mu_n = 1/\mu$), and the routing policy (LIAR policy is used). Thus, our DOE reduces to an experiment in which we seek to vary the call volume and the number of levels agents have. Holding service times constant and varying the offered load is very common in assessing queueing systems. Wallace and Whitt (2004) used this accepted practice in their resource-pooling experiments and we will use it here in our design experiment. Although the number of trunk lines and the routing policy are controllable factors, we elect to hold these two constant since call center managers are generally not as concerned with these parameters on a day-to-day operational basis. Trunk line capacity assessment are generally performed on a monthly or even quarterly basis depending on the size of

Table 2: Comparing Our SBR Simulation 95% Confidence Results against Known Models

| | | **Known** | **Simulator** |
|---|---|---|---|
| **Known Model** | **Settings** | **Results** | **Results** |
| 1. $M/M/90/0$ | $\lambda$ = 10 calls/min,<br>$1/\mu$ = 10 mins | $\epsilon \doteq 7.96\%$<br>$\upsilon \doteq 92\%$ | [7.85, 8.23]<br>[91.9, 92.2] |
| 2. Two independent<br>$M/M/C/\infty$ models:<br>$M/M/1/\infty$ and<br>$M/M/2/\infty$ | $\lambda_1$ = 0.09 calls/min,<br>$\lambda_2$ = 0.186 calls/min,<br>$1/\mu_1$ = 10 mins,<br>$1/\mu_2$ = 10 mins,<br>$\tau$ = 30 sec<br>$(C_1, C_2)$ = (1, 2) | $W \doteq 72.1$<br>$W_1$ = 90.0<br>$W_2 \doteq 63.3$<br>$\delta \doteq 89.1$ %<br>$\delta_1 \doteq 89.6\%$<br>$\delta_2 \doteq 88.9\%$<br>$\upsilon \doteq 92.0\%$ | [68.5, 73.0]<br>[80.6, 93.2]<br>[59.3, 66.6]<br>[88.7, 89.3]<br>[88.8, 90.0]<br>[88.4, 89.0]<br>[91.7, 92.1] |
| 3. $M/M/108/10$ | $\lambda$ = 10 calls/min,<br>$\tau$ = 30 sec,<br>$1/\mu$ = 10 mins | $\epsilon \doteq 1.3\%$<br>$W \doteq 0.094$<br>$\delta \doteq 8.1\%$<br>$\upsilon \doteq 91\%$ | [1.23, 1.41]<br>[0.088, 0.097]<br>[7.48, 8.32]<br>[91.0, 91.6] |
| 4. Ridley's Call Center<br>Simulator with 2 classes<br>and dynamic priorities<br>$M_2(t)/M_2/100/\infty/NPRP$ | $\lambda_1$ = 6 calls/min,<br>$\lambda_2$ = 3 calls/min,<br>$1/\mu_1$ = 10 mins,<br>$1/\mu_2$ = 10 mins,<br>$\tau$ = 30 sec | $\hat{W_1} \in [0.049, 0.060]$<br>$\hat{W_2} \in [0.434, 0.612]$<br>$\hat{\delta_1} \doteq 2.9\%$<br>$\hat{\delta_2} \doteq 15.5\%$ | [0.049, 0.059]<br>[0.430, 0.610]<br>[2.45, 3.15]<br>[14.41, 17.59] |
| 5. Stanford and Grassmann<br>Bilingual Call Center<br>$M_2/M/20/\infty/NPRP$ | $\lambda_1$ = 0.384 jobs/sec,<br>$\lambda_2$ = 0.256 jobs/sec,<br>$1/\mu_1 = 1/\mu_2$ = 25 sec,<br>$(C_1, C_2)$ = (12, 8) | $W_1$ = 1.43<br>$W_2$ = 9.39<br>$\upsilon$ = 80% | [1.38, 1.60]<br>[8.90, 10.29]<br>[79.6, 80.2] |

the call center, the service level targets, and the forecasted demands. The routing policy assessments are usually done on a less frequent basis than trunk line capacity assessments (e.g., semi-annually or annually).

We consider three levels for the call volume factor: Level 1 is the under load case, Level 2 is the target load case, and the Level 3 is the over load. As for the agent-skill matrix factor, we follow the same procedure in the resource-pooling experiments in Wallace and Whitt (2004). We consider $n$ different levels for this factor where $n$ is the number of call types. We consider the case where agents have only one skill described in the matrix in $A^{(1)}_{C\times n}$, agents have two skills described in $A^{(2)}_{C\times n}$, and so on, up to the case where agents have all $n$ skills described $A^{(n)}_{C\times n}$. *Full resource pooling* exists when all agents have all $n$ skills. Thus, our statistical analysis using the $n+1$ potential factors is reduced to a $3^n \times n$ design experiment assuming a single replicate. In our model, a single repetition of the basic experiment is performed with each execution or run of our SBR call center simulation. Thus, in a single replication we have $3^n \times n$ observations upon which to obtain estimates for the experimental error. If the number of call types $n$ is different from 3 (i.e., $n \neq 3$) then we have a mixed-factorial design experiment.

### 4.1 $3^4 \times 4$ Mixed Factorial Design

We consider a SBR call center design experiment that has an environment with 4 different service requests and 4 different work groups. We fix the number of agents $C$, trunklines $C + K$, and mean service times ($1/\mu_1 = \ldots = 1/\mu_n = 1/\mu$). We use the longest-idle-agent routing (LIAR) policy to route calls to agents. These values along with the three levels per call rate factor for our SBR call center transform our analysis to an $3^4 \times 4$ mixed-factorial-design experiment. In this case, we will have 5 ($n + 1 = 4 + 1 = 5$) factors of which 4 are input factors that represent the 4 different call rates, each with 3 different levels and one is the agent skill matrix factor. The agent skill matrix factor has 4 levels. The first level is the case in which all agents have only one skill and the fourth level is the case in which all agents have all 4 skills.

### 4.2 Assumptions

We consider numerical examples using a $M_4/M/60/10/NPRP$ SBR call center configured with the following assumptions:

1. Fixed Assumptions - $C$ = 60 agents, $n$ = 4 call types, $m$ = 4 work groups, $C_1 = \cdots = C_4 = 15$ agents per work group, target service level response $t$ = 0.5 minutes, call volume are equal ($\lambda_1 = \cdots = \lambda_4$), and secondary skills are evenly distributed

across work group agents. Fixed factors consist of $C + K = 70$ telephone lines, 10 minutes mean service times ($1/\mu_1 = \cdots = 1/\mu_4 = 10$), and LIAR policy to route calls.

2. Variable (Factor) Assumptions - Number of skills agents have, factor E, will vary from 1 (Level 1) to 4 (Level 4) as predefined by the agent skill matrix $A^{(s)}_{60\times 4}$ ($s = 1, \ldots, 4$) Aggregate offered load ($\rho = \lambda/\mu$) is varies from to 48.0 (Level 1 or under load), 54.0 (Level 2 or target load) and 66.0 (Level 3 or over load).

The agent skill matrices used are as follows:

$$\mathbf{A}^{(1)}_{60\times 4} = \begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0
\end{pmatrix}, \mathbf{A}^{(2)}_{60\times 4} = \begin{pmatrix}
1 & 2 & 0 & 0 \\
1 & 2 & 0 & 0 \\
1 & 2 & 0 & 0 \\
1 & 2 & 0 & 0 \\
1 & 2 & 0 & 0 \\
1 & 3 & 0 & 0 \\
1 & 3 & 0 & 0 \\
1 & 3 & 0 & 0 \\
1 & 3 & 0 & 0 \\
1 & 3 & 0 & 0 \\
1 & 4 & 0 & 0 \\
1 & 4 & 0 & 0 \\
1 & 4 & 0 & 0 \\
1 & 4 & 0 & 0 \\
1 & 4 & 0 & 0 \\
2 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 \\
2 & 3 & 0 & 0 \\
2 & 3 & 0 & 0 \\
2 & 3 & 0 & 0 \\
2 & 3 & 0 & 0 \\
2 & 3 & 0 & 0 \\
2 & 4 & 0 & 0 \\
2 & 4 & 0 & 0 \\
2 & 4 & 0 & 0 \\
2 & 4 & 0 & 0 \\
2 & 4 & 0 & 0 \\
3 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 \\
3 & 2 & 0 & 0 \\
3 & 2 & 0 & 0 \\
3 & 2 & 0 & 0 \\
3 & 2 & 0 & 0 \\
3 & 2 & 0 & 0 \\
3 & 4 & 0 & 0 \\
3 & 4 & 0 & 0 \\
3 & 4 & 0 & 0 \\
3 & 4 & 0 & 0 \\
3 & 4 & 0 & 0 \\
4 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 \\
4 & 2 & 0 & 0 \\
4 & 2 & 0 & 0 \\
4 & 2 & 0 & 0 \\
4 & 2 & 0 & 0 \\
4 & 2 & 0 & 0 \\
4 & 3 & 0 & 0 \\
4 & 3 & 0 & 0 \\
4 & 3 & 0 & 0 \\
4 & 3 & 0 & 0 \\
4 & 3 & 0 & 0
\end{pmatrix}.$$

$$\mathbf{A}^{(3)}_{60\times 4} = \begin{pmatrix}
1 & 2 & 3 & 0 \\
1 & 2 & 3 & 0 \\
1 & 2 & 3 & 0 \\
1 & 2 & 3 & 0 \\
1 & 2 & 3 & 0 \\
1 & 3 & 4 & 0 \\
1 & 3 & 4 & 0 \\
1 & 3 & 4 & 0 \\
1 & 3 & 4 & 0 \\
1 & 3 & 4 & 0 \\
1 & 4 & 2 & 0 \\
1 & 4 & 2 & 0 \\
1 & 4 & 2 & 0 \\
1 & 4 & 2 & 0 \\
1 & 4 & 2 & 0 \\
2 & 1 & 3 & 0 \\
2 & 1 & 3 & 0 \\
2 & 1 & 3 & 0 \\
2 & 1 & 3 & 0 \\
2 & 1 & 3 & 0 \\
2 & 3 & 4 & 0 \\
2 & 3 & 4 & 0 \\
2 & 3 & 4 & 0 \\
2 & 3 & 4 & 0 \\
2 & 3 & 4 & 0 \\
2 & 4 & 1 & 0 \\
2 & 4 & 1 & 0 \\
2 & 4 & 1 & 0 \\
2 & 4 & 1 & 0 \\
2 & 4 & 1 & 0 \\
3 & 1 & 2 & 0 \\
3 & 1 & 2 & 0 \\
3 & 1 & 2 & 0 \\
3 & 1 & 2 & 0 \\
3 & 1 & 2 & 0 \\
3 & 2 & 4 & 0 \\
3 & 2 & 4 & 0 \\
3 & 2 & 4 & 0 \\
3 & 2 & 4 & 0 \\
3 & 2 & 4 & 0 \\
3 & 4 & 1 & 0 \\
3 & 4 & 1 & 0 \\
3 & 4 & 1 & 0 \\
3 & 4 & 1 & 0 \\
3 & 4 & 1 & 0 \\
4 & 1 & 2 & 0 \\
4 & 1 & 2 & 0 \\
4 & 1 & 2 & 0 \\
4 & 1 & 2 & 0 \\
4 & 1 & 2 & 0 \\
4 & 2 & 3 & 0 \\
4 & 2 & 3 & 0 \\
4 & 2 & 3 & 0 \\
4 & 2 & 3 & 0 \\
4 & 2 & 3 & 0 \\
4 & 3 & 1 & 0 \\
4 & 3 & 1 & 0 \\
4 & 3 & 1 & 0 \\
4 & 3 & 1 & 0 \\
4 & 3 & 1 & 0
\end{pmatrix}, \mathbf{A}^{(4)}_{60\times 4} = \begin{pmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
1 & 3 & 4 & 2 \\
1 & 3 & 4 & 2 \\
1 & 3 & 4 & 2 \\
1 & 3 & 4 & 2 \\
1 & 3 & 4 & 2 \\
1 & 4 & 2 & 3 \\
1 & 4 & 2 & 3 \\
1 & 4 & 2 & 3 \\
1 & 4 & 2 & 3 \\
1 & 4 & 2 & 3 \\
2 & 1 & 3 & 4 \\
2 & 1 & 3 & 4 \\
2 & 1 & 3 & 4 \\
2 & 1 & 3 & 4 \\
2 & 1 & 3 & 4 \\
2 & 3 & 4 & 1 \\
2 & 3 & 4 & 1 \\
2 & 3 & 4 & 1 \\
2 & 3 & 4 & 1 \\
2 & 3 & 4 & 1 \\
2 & 4 & 1 & 3 \\
2 & 4 & 1 & 3 \\
2 & 4 & 1 & 3 \\
2 & 4 & 1 & 3 \\
2 & 4 & 1 & 3 \\
3 & 1 & 2 & 4 \\
3 & 1 & 2 & 4 \\
3 & 1 & 2 & 4 \\
3 & 1 & 2 & 4 \\
3 & 1 & 2 & 4 \\
3 & 2 & 4 & 1 \\
3 & 2 & 4 & 1 \\
3 & 2 & 4 & 1 \\
3 & 2 & 4 & 1 \\
3 & 2 & 4 & 1 \\
3 & 4 & 1 & 2 \\
3 & 4 & 1 & 2 \\
3 & 4 & 1 & 2 \\
3 & 4 & 1 & 2 \\
3 & 4 & 1 & 2 \\
4 & 1 & 2 & 3 \\
4 & 1 & 2 & 3 \\
4 & 1 & 2 & 3 \\
4 & 1 & 2 & 3 \\
4 & 1 & 2 & 3 \\
4 & 2 & 3 & 1 \\
4 & 2 & 3 & 1 \\
4 & 2 & 3 & 1 \\
4 & 2 & 3 & 1 \\
4 & 2 & 3 & 1 \\
4 & 3 & 1 & 2 \\
4 & 3 & 1 & 2 \\
4 & 3 & 1 & 2 \\
4 & 3 & 1 & 2 \\
4 & 3 & 1 & 2
\end{pmatrix}.$$

## 5 NUMERICAL RESULTS

Design-Ease statistical software package is used to calculate the statistical interactions between the key variables. See `<http://www.statease.com/>` and Anderson and Whitcomb (2000) for more details on Design-Ease. In this section, we focus our attention on understanding the performance of the SBR call center environment under study. We seek to understand what happens to the performance

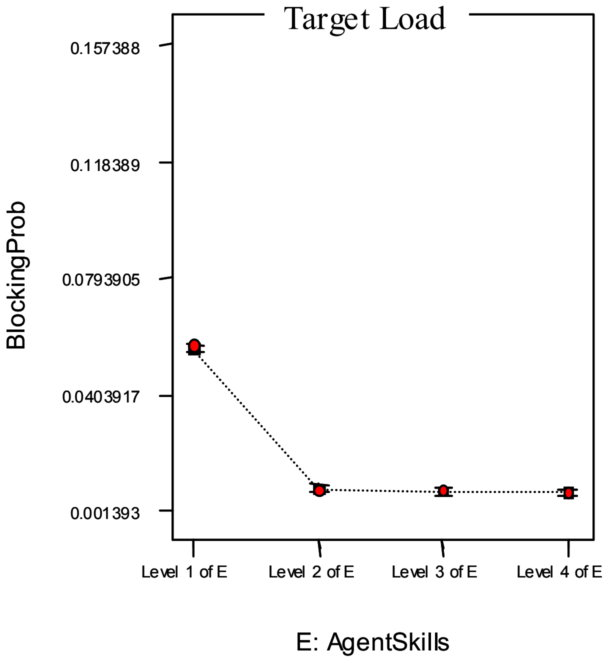

Figure 2: One-Factor Plot for Blocking that illustrates Resource Pooling.

measures as we vary key factors like the different call volume and the number of skills agents have.

While the Design-Ease package produces many useful statistical results (e.g., ANOVA and regression models), we focus our attention on the tools ability to illustrate factor interactions. Also, we seek to validate the resource-pooling phenomenon observed in Wallace and Whitt (2004). The DOE resulted in a model for which all input parameters selected (independent variables) were significant at the 0.0001 level of significance. This is not surprising as the input parameters were selected based on known queueing results.

We illustrate resource-pooling phenomenon using several different graphs from Design-Ease. Figure 2 shows a simple single plot graph for blocking under target load which is comparable to the one-factor-at-a-time experiments conducted in Wallace and Whitt (2004). We see that the system where agents have two skills (Level 2 of E) achieves nearly the same performance level as the system where agents have all four skills (Level 4 of E). This resource-pooling behavior is observed across all offered loads and performance measures.

Using single plots, we also observe known queueing results in our DOE output (e.g., increase in offered load increases both blocking and delay). As the number of skills agents have increases and call rates are fixed, we make the following observations. The blocking and delay performance measures decrease. The percent of calls answered with $\tau$ minutes increases. Also, the both agents utilization and work group utilization increase which support the claim that skill-based routing improves agents and work group productivity. On the other hand, agents primary utilization decreases when the number of skills agents have increases.

Factor interaction is a more distinguishing feature of factorial experiments that is not possible with one-factor-at-a-time experiments. An interaction is the failure of the one factor to produce the same effect on the response at different levels of another factor. Graphically, strong interaction between two factors exhibits nonparallel lines while parallel lines show no interaction.

Using Design-Ease, we describe two-factor interactions using three plots as shown in Figures 3 and 4. Both figures illustrate two-factor interaction and depict the other graphs used to show resource pooling. The graphs where agents have two skills, plot (b), are very similar to the graphs where agents have all four skills, plot (c). This is another confirmation of resource pooling.

In addition, Figures 3 and 4 show that there are no interaction between the two call types when agents have two skills or more. This remarkable observation shows no interaction exists under resource pooling. In all cases, we observe parallel lines in two-factor interaction when the agents have two or more skills.

## 6 CONCLUSIONS

Simulation-based analysis of skill-based routing call centers is expected to be the predominant tool of choice since even for relatively simple systems, available analytical solutions are rather restricted. See Garnett and Mandelbaum (2001). The presented simulation model developed proved to be a flexible tool in analyzing the SBR call center problem. The combination of simulation with the design of experiment framework provided not only insights to the relationship between model inputs and key performance measures but also the interactions between them. Traditional one-factor-at-a-time experiments are not able to capture these critical interactions.

## ACKNOWLEDGMENTS

This paper is based on the second author's doctoral dissertation at The George Washington University, co-advised by William A. Massey from Princeton University, Ward Whitt from Columbia University, and the first author, Thomas A. Mazzuchi from The George Washington University. The authors are grateful to the other co-advisors for their assistance.

## REFERENCES

Anderson, M. J. and P. J. Whitcomb. 2000. *DOE Simplified, Practical Tools for Effective Experimentation*. Productivity, Inc.

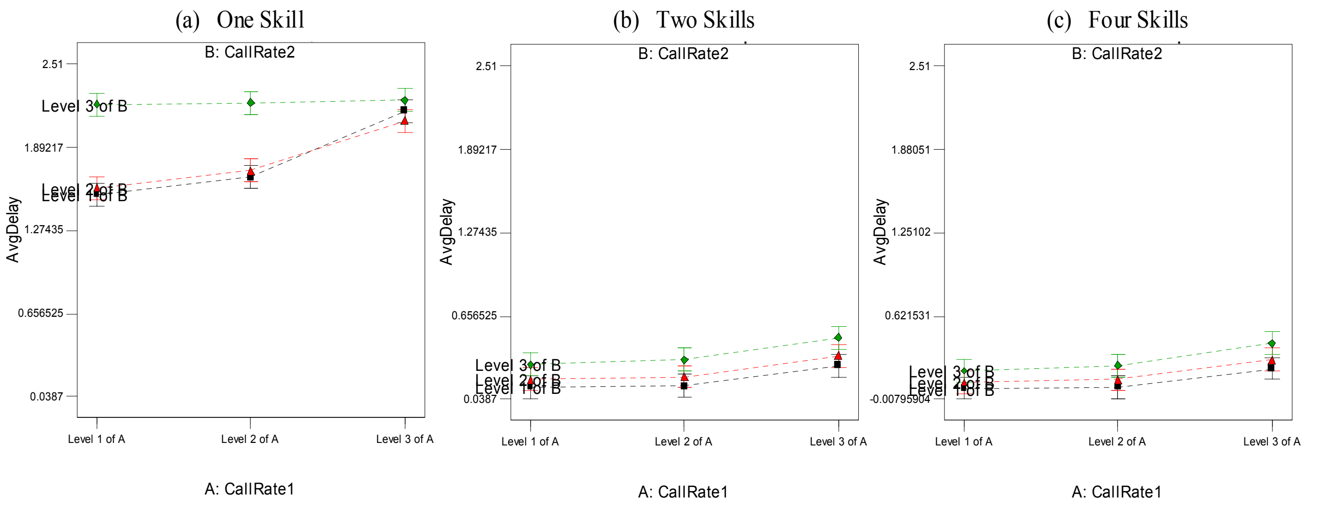

Figure 3: Interaction graphs between Call Type 1, Call Type 2 and Number of Skills per Agent for the Average Delay with Call Type 3 and Call Type 4 are both at Level 2.

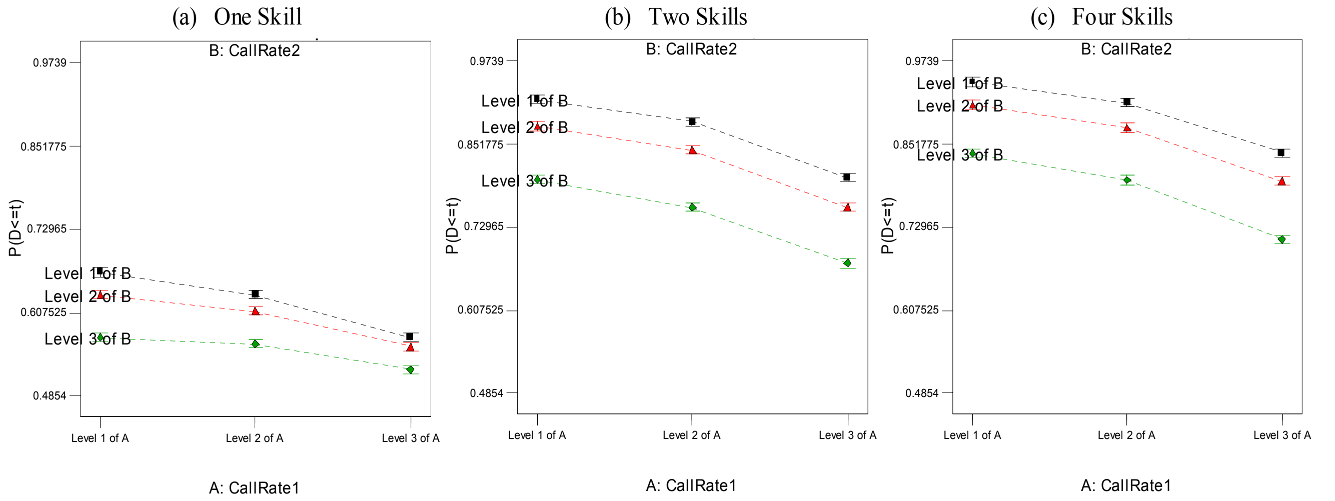

Figure 4: Interaction graphs between Call Type 1, Call Type 2 and Number of Skills per Agent for the Probability of Delay with Call Type 3 and Call Type 4 are both at Level 2.

Gans, N., G. Koole, and A. Mandelbaum. 2003. Telephone Call Centers: Tutorial, Review and Research Prospects. *Manufacturing and Service Operations Management*. 5: 79–141.

Garnett, O. and A. Mandelbaum. 2001. An Introduction to Skills-Based Routing and its Operational Complexities. Working Notes, Technion.

Gross, D. and C. Harris. 1998. *Fundamentals of Queueing Theory*, 3rd edition, John Wiley & Sons.

Harris, C., K. Hoffman, and P. Saunders. 1987. Modeling The IRS Telephone Taxpayer Information System. *Operations Research*. 35: 504–523.

Jain, R. 1991. *The Art of Computer Systems Performance Analysis , Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons.

Law, A. and W. Kelton. 2001. *Simulation Modeling and Analysis*, 3rd edition, McGraw Hill.

Massey, W. A. and R. B. Wallace. 2004. An Optimal Design of the M/M/C/K Queue for Call Centers. To appear in *Queueing Systems: Theory and Application*.

Montgomery, D. C. 2001. *Design and Analysis of Experiments*, 5th edition, John Wiley & Sons.

Ridley, A. 2004. Performance Analysis of a Multi-Class Preemptive Priority Call Center with Time Varying

Arrivals, Ph.D. dissertation, University of Maryland, College Park.

Ross, S. 2002. *Simulation*, 3rd edition, Academic Press.

Stanford, D. and W. K. Grassmann. 1998. Bilingual server call centers. *Analysis of Communication Networks: call centers, traffic and performance*, eds. D. McDonald and S. R. E. Turner, 31–47, Providence: *American Mathematics Society*.

Wallace, R. B. 2004. Performance Modeling and Design of Call Centers with Skill-Based Routing, D.Sc. Dissertation, The George Washington University.

Wallace. R. B. and W. Whitt. 2004. Resource Pooling and Staffing in Call Centers with Skill-Based Routing. Submitted to *Operations Research*.

Whitt, W. 1989. Planning Queueing Simulations. *Management Science* 35 (11): 1341–1366.

## AUTHOR BIOGRAPHIES

**THOMAS A. MAZZUCHI** is professor and Chair of the Department of Engineering Management and Systems Engineering at The George Washington University. His research interests include reliability and risk analysis, Bayesian inference and quality control. He can be contacted by e-mail at <mazzu@gwu.edu>.

**RODNEY B. WALLACE** is a Technical Solutions Manager at IBM Global Services and has a doctoral degree in Operations Research from The George Washington University. His research interests include computer systems performance analysis, queueing theory and simulation. He is a member of INFORMS and his e-mail address is <rodney.wallace@us.ibm.com>.