Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

USING SIMULATION-BASED STOCHASTIC APPROXIMATION TO OPTIMIZE STAFFING OF SYSTEMS WITH SKILLS-BASED-ROUTING

Zohar Feldman

Haifa University Campus
IBM Haifa Research Labs
Haifa 31905, Israel

Avishai Mandelbaum

Industrial Engineering and Management
Technion Institute of Technology
Haifa 32000, Israel

ABSTRACT

In this paper, we consider the problem of minimizing the operational costs of systems with *Skills-Based-Routing* (SBR). In such systems, customers of multiple classes are routed to servers of multiple skills. In the settings we consider, each server skill is associated with a corresponding cost, and service level can either appear as a strong constraint or incur a cost.
The solution we propose is based on the *Stochastic Approximation* (SA) approach. Since SBR models are analytically intractable in general, we use computer simulation to evaluate service-level measures. Under the assumption of convexity of the service-level as functions in staffing levels, SA provides an analytical proof of convergence, together with a rate of convergence. We show, via numerical examples, that although the convexity assumption does not hold for all cases and all types of service-level objectives, the algorithm nevertheless identifies the optimal solution.

1 INTRODUCTION

This paper is motivated mainly by Telephone Call Centers, where the term SBR is originating from. The SBR framework, however, arises in other contexts of service systems, for example technical support providers, banking systems, health-care etc. In these settings, heterogenous customers are routed to appropriately skilled server according to some custom routing policy.
Call centers, and service systems in general, give rise to many operational problems. A central example is the *Staffing Problem*: under an existing operational reality, finding the minimal-cost staffing levels that is required to meet some given *Quality of Service* (QoS) constraints. This problem has received a great deal of attention over the years, as it rightly deserves: staffing costs are estimated at about 70% of a call center's operational costs. Staffing "wisely" can thus result in substantial savings while achieving operational objectives.
We define two different optimization formulations. In one, we find the minimal cost staffing levels whereas service level appears as a hard constraint that must be satisfied. In the other formulation, service levels incur costs; we thus try to identify the staffing levels that minimize the total operational costs.
Due to model complexities, there exists only limited amount of analytical results for very special cases of SBR designs. Some of these works include Gurvich, Armony, and Mandelbaum (2008) and Armony and Mandelbaum (2004) who identify an asymptotical optimal routing and staffing schemes for the $\vee$-design and the inverted-$\vee$ design respectively. In the $\vee$-design, there are several customer classes and all servers are statistically identical, whereas in the inverted-$\vee$ design (also denoted $\wedge$-design), there is a single customer class and several server skills. Gurvich and Whitt (see Gurvich and Whitt 2010 and Gurvich and Whitt 2008), propose a family of Routing controls called Queue-Idleness-Ratio (QIR) rules which, under some regularity conditions on the network structure, facilitates the establishment of a solution to the joint problem of design, staffing and routing in a nearly optimal manner.
Another approach is to use simplifying analytical models to propose an initial intelligent solution, then apply simulation in order to validate and refine this solution (cf. Wallace and Whitt 2005). Although this approach is generally heuristic, it

typically produces good results for a broad range of problems, and in some cases it even provides important insights.
In this paper, we use simulation to evaluate service levels. The main advantage of using simulation is its ability to accommodate complex models that capture reality far better than simplifying analytical models. Complicated features such as time-varying rates, general distributions and even uncertainty of the parameters value can be integrated into the simulation fairly easily. On the other hand, simulation by itself does not provide any closed-form expression nor a prescription to identify the optimal solution as can be often derived directly by analytical models. However, simulation is an integral part of a certain class of stochastic optimization computational approach which is based on Monte-Carlo sampling techniques.
The most common approach based on Monte-Carlo sampling techniques is the Sample Average Approximation (SAA). SAA approximates the original stochastic problem by a sample average problem, in which the "true" function is replaced with a sample average function. The obtained SAA problem, however, still has to be solved by an appropriate numerical method. The SAA was used in many contexts under different names, and specifically in the context of Call Centers staffing. Atlason, Epelman, and Henderson (2004) present a simulation-based analytic center cutting plane method to solve a sample average approximation of the problem of minimizing staffing costs in an inbound call center, while maintaining desirable service level in multiple time periods. A similar approach is taken in Cezik and L'ecuyer (2008) for minimizing the staffing costs of a multiskill call center. Namely, Cezik and L'ecuyer (2008) uses an iterative cutting-plane algorithm on an integer program, with additional heuristics to overcome numerical difficulties.
We use an alternative approach called the Stochastic Approximation (SA). SA goes back to the pioneering paper by Robbins and Monro 1951, and has been significantly improved and modified by various works. Some of the important contributions appear in Polyak and Juditsky 1992 and a recent work by Juditsky et al. 2008. Unlike SAA, SA does not attempt to provide statistically significant estimations for the inspected solutions, rather it uses noisy estimation to iterate between solutions according to subgradient descent methods. The convergence to global optima thus depends on the convexity of the stochastic function. Similarly to Atlason, Epelman, and Henderson (2004) and Cezik and L'ecuyer (2008), we make the assumption that the service level functions that we consider are convex in the staffing levels.
Although combined with a good algorithm, SAA was traditionally considered to produce superior performance over the SA method, it was demonstrated in Juditsky et al. 2008 that their modified SA approach can be competitive and even significantly outperform the SAA for a certain class of stochastic problems.
The rest of this paper is organized as follows. In §2 we introduce our model for an SBR service system and formulate corresponding optimal staffing problems. We also describe in general the SA approach with some modified and improved procedures that we use in our algorithms. We describe in details our proposed SA-based solution in §3, presenting our algorithms and sub-routines. We also provide a proof of convergence stemming from Stochastic Approximation theory. In §4 we describe a numerical study and demonstrate the accuracy of our algorithm in various settings of SBR systems and routing controls and also in time-varying environments. Our experiments show that the convexity assumption is applicable in many cases and, more importantly, even when the convexity assumption is refuted, our algorithms still provide optimal solutions.
Finally, §5 concludes this paper and propose future work.

2 PROBLEM FORMULATION

The model we consider, in its most general form is described as follows: there is a finite $\mathscr{I}$ of customer classes, and a finite set $\mathscr{J}$ of server pools (stations). We denote the cardinality of the sets $\mathscr{I}$ and $\mathscr{J}$ by I and J, respectively. Customers of class $i \in \mathscr{I}$ arrive according to a Poisson process with rate λ_i (note that in the time-varying context λ_i may be a function of time), and the time they are willing to wait before abandoning is taken from a distribution $\mathscr{D}_i^P$ (the superscript P stands for Patience). Servers of pool $j \in \mathscr{J}$ process customers of class $i \in \mathscr{I}$ according to a service duration from distribution $\mathscr{D}_{i,j}^S$ (the superscript S stands for Service). Moreover, each pool of servers is associated with a cost such that the staffing cost per time unit of a server from pool $j \in \mathscr{J}$ is c_j.
We consider two optimization problems: a **Cost Optimization Problem** and a **Constraint Satisfaction Problem**.
The **Cost Optimization Problem** is formulated as follows:

$$\begin{aligned} \min_{N} \quad & c^T N + \sum_{k=1}^{K} f^k(N) \\ \text{s.t.} \quad & A^T N \leq b \\ & N \in \mathbb{Z}_+^J \end{aligned} \tag{1}$$

Here $f^k(N)$ are service-level related average cost functions, where cost is measured per unit of time. For instance, $f^k(N) = c_k^{ab}\lambda_k \mathbb{P}_N\{ab_k\}$ is the average cost of abandonments per unit of time, in case each abandonment of class k customer incurs a cost of c_k^{ab}. Another example for such a function may be $f^k(N) = \lambda_k \mathbb{E}_N\left\{c_k^W(W_k)\right\}$, which is the expected cost of waiting time per time unit, given a cost function c_k^W for the waiting-time-in-queue of class k customers.

In the **Constraint Satisfaction Problem**, one is interested in solving

$$\begin{array}{ll} \min\limits_{N} & c^T N \\ \text{s.t.} & f^k(N) \leq \alpha_k \quad \forall k = 1,\ldots,K \\ & A^T N \leq b \\ & N \in \mathbb{Z}_+^J \end{array} . \tag{2}$$

Here $f^k(N)$ denotes a *Service Level Objective* associated with a specific performance measure and class of customers. The most common objective takes the form "no more than 20% of type i customers are to wait more than 20 seconds" which can be expressed mathematically by $\mathbb{P}\{W_i > 20sec\} \leq 0.2$. Other typical constraints include performance measures such as the probability of abandonment ($\mathbb{P}\{Abandon\}$) or average wait ($\mathbb{E}\left\{W_q\right\}$). An important performance measure, rarely tracked, is the delay probability ($\mathbb{P}\left\{W_q > 0\right\}$).
In both problem formulations (1) and (2), we are allowing additional linear constraints on the staffing levels. These constraints arise from operational issues such as servers' availability, training considerations etc. For instance, if there are only 50 expert servers, one should incorporate the simple constraint $N_{expert} \leq 50$. The requirement that at least 10% out of the entire workforce should be trainees is translated into the linear constraint $N_{trainee} \geq 0.1\left(N_{trainee} + N_{expert}\right)$, and so on. More importantly, we use these linear constraints to introduce Stability Conditions: in a multi-class, multi-pool system, this is the set of necessary conditions for stabilizing the system, i.e. none of the queues explodes in the long run. The contribution of these constraints is two-fold: first, the search space is reduced and thus convergence is faster; second, the service level functions, reduced to the restricted region, are more likely to be convex and thus the validity of the algorithm is guaranteed. Indeed, if we look at a single-class single-pool queue (G/G/N), then service level measurements such as the delay probability are convex on the set $N \geq R \triangleq \frac{\lambda}{\mu}$, but not on $\mathbb{Z}_+^J$. Namely, the delay probability is essentially 1 for all staffing levels that are smaller than the offered load R, and start decreasing in a convex manner from that point and on. In our multi-class, multi-pool situation, we expect to have the same behavior component-wise, i.e. if we examine the staffing levels in each pool while fixing the staffing in all others, the delay probability of a certain class will equal to one for any staffing levels that falls out of the "stability" region, and decrease monotonically when increasing the staffing in the specific pool in the "stability" region.

The main difficulty of solving (1) and (2) is the intractability of the functions f^k. Indeed, only for a scanty set of relatively simple models, with a specific design and control, few objective functions similar to the ones described above can be calculated either analytically, numerically or asymptotically. One is then led naturally to resort to simulation.

The solution we propose is based on a *Stochastic Optimization* approach called *Stochastic Approximation* (SA). We give a general outline of this approach and demonstrate its application to our problem in the following section.

3 STOCHASTIC APPROXIMATION

The *Stochastic Approximation* (SA) approach uses *Monte-Carlo* sampling techniques to solve optimization problems of the form

$$\min_{x \in X}\left\{f(x) := \mathbb{E}\left\{F(x,\xi)\right\}\right\}, \tag{3}$$

where $X \subset \mathbb{R}^n$ is a non-empty bounded closed convex set, ξ is a random vector whose probability distribution $\mathbf{P}$ is supported on a set $\Xi \subset \mathbb{R}^d$, and $F : X \times \Xi \to \mathbb{R}$. It is further assumed that, for every $\xi \in \Xi$, the function $F(\cdot,\xi)$ is convex and that, for every $x \in X$, the function $F(x,\cdot)$ is integrable with respect to $\mathbf{P}$.
A recent work by Juditsky et al. 2008 extended this framework to solve also problems that take the form

$$\min_{x \in X} \max_{k=1,\ldots,K}\left\{f^k(x) := \mathbb{E}\left\{F^k(x,\xi)\right\}\right\}. \tag{4}$$

SA makes the following assumptions:

Assumption 1. *There is a way of generating iid samples $\xi_1, \xi_2, \ldots$ of the random vector ξ.*

Assumption 2. *There is an Oracle at our disposal that, for any $\xi \in \Xi$ and $x \in X$, returns the value of $F(x,\xi)$ and a stochastic sub-gradient - a vector $\mathrm{G}(x,\xi)$ such that $g(x) := \mathbb{E}\{\mathrm{G}(x,\xi)\}$ is well defined and is a sub-gradient of $f(\cdot)$ at x, i.e. $f(y) \geq f(x) + g^T(x)(y-x), \forall y \in X$ $(g(x) \in \partial f(x))$.*

Before presenting our SA-based optimization algorithms, let us first establish the connection between the objective functions in (3) and (4) considered by the SA and the service level functions $f^k(N)$ from our optimization problems (1) and (2).

Denote by $(\Omega, \mathscr{F}, P)$ the probability space formed by the distributions of arrival times, service times and patience. For some SL types, the expectation form arises quite naturally. For instance, if we look at SL functions of the form $f(N) = \mathbb{E}[\#Abandons]$, i.e. the expected number of abandonments, over a given optimization horizon and staffing levels $N \in \mathbb{R}^n$, then we fall exactly into the domain of SA. Namely, define $F : \mathbb{R}^J \times \Omega \to \mathbb{R}$ such that $F(N,\omega)$ is a random variable denoting the number of abandonments, given a possible outcome ω and staffing levels $N \in \mathbb{Z}_+^J$. We can write

$$f(N) := \int_\Omega F(N,\omega)\, dP(\omega), \tag{5}$$

which is equivalent to $f(x) := \mathbb{E}[F(x,\xi)]$ in (3): in (3) the random variable $F(x,\omega)$ is defined with an intermediate random vector ξ, where in our case it is imbedded in the probability space $(\Omega, \mathscr{F}, P)$.

In other cases, however, we may need to perform additional steps before using SA. For example, when using simulation, the conventional way to represent the SL function $\mathbb{P}\{W_q > 0\}$ is by the following ratio:

$$\mathbb{P}_N\{W_q > 0\} = \frac{\int_\Omega D(N,\omega)\, dP(\omega)}{\int_\Omega A(\omega)\, dP(\omega)}; \tag{6}$$

here $D(N,\omega)$ denotes the number of delayed customers and $A(\omega)$ denotes the total number of arrivals, over a given time-horizon.

Fortunately, the expectation in the denominator of (6) is independent of the staffing levels N. We can thus accommodate such SL functions as follows: In Cost Optimization, f appears in the objective function. Since the denominator is independent of the decision variables N, we can ignore it completely as it has no affect over the solution. In this case we just set $F(N,\omega) := D(N,\omega)$.

In the Constraint Satisfaction, the SL functions appear as constraints, e.g.

$$\mathbb{P}\{W > 0\} \leq \alpha \tag{7}$$

In this case, we translate (7) into

$$\mathbb{E}[D(N,\cdot)] \leq \alpha \mathbb{E}[A(\cdot)], \tag{8}$$

where $\mathbb{E}[A(\cdot)]$ can be calculated in advance, and therefore the right-hand-side of (8) is constant.

We use computer simulation to both generate iid sample $\omega_1, \omega_2, \ldots$ from Ω, and calculate the values of $F^k(N,\omega)$ for any $N \in \mathbb{R}_+^J$ and $\omega \in \Omega$ that arise along the steps of the algorithms. An important point to be made here is that, in practice, we are required to evaluate the functions $F^k(N,\omega)$ for non-integral values of N, even though N is discrete in our case (there is no such thing as a fraction of a server). In this case, we round the point N to the nearest integral point and plug it into the simulation to obtain the required value.

Similarly to (Atlason, Epelman, and Henderson 2004), we evaluate the stochastic sub-gradient $\mathrm{G}^k(N,\omega)$ by taking finite differences of length 1 as follows:

$$\left\{\mathrm{G}^k(N,\omega)\right\}_i := F^k(N+e_i,\omega) - F^k(N,\omega). \tag{9}$$

Here, e_i denotes a vector of the same dimension as N, in which all entries are zeros except for the i'th component which equals to one.
There are several optional ways of producing sub-gradients, such as infinitesimal perturbation analysis (cf. Ho and Cao 1983) and likelihood ratio gradient estimation (cf. Glynn 1987). In fact, the finite differences method does not guarantee to yield a sub-gradient. Examples in which this method fails are easy to find. Nevertheless, finite differences still appear as a reasonable method and our numerical studies demonstrate that the method could yield optimal solutions.

3.1 Cost Optimization

We apply the *Robust SA* algorithm in a straightforward manner to solve the Cost Optimization Problem.
The Robust SA algorithm solves (3) by mimicking a simple gradient descent method and consequently averaging the obtained iterates (cf. Polyak and Juditsky 1992, Juditsky et al. 2008). More specifically, given a step size γ and the number of iterations T, the Robust SA algorithm may be defined by the following iterative step:

$$x_{j+1} := \Pi_X\left(x_j - \gamma G\left(x_j, \xi_j\right)\right), j = 1, 2, \ldots, \tag{10}$$

and it produces $\hat{x}_T$ as a *final solution*:

$$\hat{x}_T := \frac{1}{T}\sum_{t=1}^{T} x_t, \tag{11}$$

where $\Pi_X(x)$ is the metric projection operator onto the set X, that is $\Pi_X(x) := \arg\min_{x' \in X} ||x' - x||$. We will use this notation throughout this paper.
A few preliminary calculations are required in order to provide the step size γ and the number of iteration T, which guarantee a desirable accuracy of the solution.
These include, among other quantities, to calculate the bound $M_*^2 := \sup_{x \in X} \mathbb{E}\left\{||G(x,\cdot)||_2^2\right\}$, where G is the sub-gradient of the objective function, composed of the service cost vector and the summation of sub-gradients of the SL cost functions. Formally, $G(x,\omega) := c + \sum_{k=1}^{K} G^k(x,\omega)$. In most cases, finding a bound to the subgradient G^k is not a difficult task. Such bounds could be obtained, for instance, using simple queueing models or simulation.
We define the search space $X = \left\{x : A^T x \leq b\right\}$. For practical considerations, we further reduce X by intersecting it with the J-dimensional cube $\mathscr{C} = \left\{x : 0 \leq x_j \leq x_j^b, \quad \forall j = 1, \ldots, J\right\}$. Here, x_j^b is the required number of agents in pool j that guarantees a perfect service experience (delay probability of all the customer classes that are served by this pool is essentially 0), given that all customers that could are served by this pool only. We compute the value of x_j^b by finding the minimal staffing level in a corresponding Erlang model for which the delay probability is 0 (practically, smaller than some small threshold) with the cumulative arrival rate $\lambda = \sum_{i \in I(j)} \lambda_i$ and service rate $\mu = \frac{1}{\lambda}\sum_{i \in I(j)} \frac{\lambda_i}{\mu_{ji}}$; $I(j)$ denotes the set of classes that are served by pool j. It is easy to see that any solution that is not in $\mathscr{C}$ is dominated in terms of cost by some solution in $\mathscr{C}$. Note that the additional constraints imposed by $\mathscr{C}$ are useful in reducing the search space, which eventually decreases the number of required iterations.

Algorithm: Cost Optimization(δ, ε)

Initialization: Calculate γ, T and x^b according to 3.1. Set $i := 1$; $x_i := \Pi_X\left(\frac{1}{2}x^b\right)$

Step 1 Run simulation to generate ω_i and obtain the value $G^k(x_i, \omega_i)\ \forall k = 1, \ldots, K$;

Step 2 Set $G(x_i, \omega_i) := c + \sum_{k=1}^{K} G^k(x_i, \omega_i)$, $x_{i+1} := \Pi_X\left(x_i - \gamma G(x_i, \omega_i)\right)$;

Step 3 If $i \geq T$ return the solution $\hat{x}_T := \frac{1}{T}\sum_{t=1}^{T} x_t$;

Step 4 Set $i := i + 1$. Go to Step 1;

Figure 1: Cost Optimization Algorithm

The Cost Optimization algorithm is displayed in Figure 1.

Theorem 1. *Upon termination of the Cost Optimization Algorithm at point $\hat{x}_T$, with step-size γ, we achieve*

$$\mathbb{P}\{f(\hat{x}_T) - f(\bar{x}) \geq \delta\} \leq \varepsilon, \tag{12}$$

where f is the objective function in the Cost Optimization Problem (1)*, and $\bar{x}$ is a minimizer of f.*

Proof. It follows from (2.19) in Juditsky et al. 2008 that $\mathbb{E}\{f(\hat{x}_T) - f(\bar{x})\} \leq \varepsilon\delta$. Then (12) is immediate by the Markov inequality. □

3.1.1 Constraints Satisfaction

The Constraint Satisfaction Problem appears to be more complicated in the sense that it does not fit the form of the optimization problem (3). Namely the functions $f^k(x)$ appear as constraints rather than objective functions. To this end, we start first with solving the following Minimax Problem

$$\min_{x \in X_C} \max_{k=1,\ldots,K} \left\{ f^k(x) - \alpha_k \right\}, \tag{13}$$

where $X_C := \left\{x \in X : c^T x = C\right\}$. Suppose that solving (13) yields some solution $\hat{x}$. We shall then be able to identify, with the help of some additional validation mechanism, if $\hat{x}$ satisfies the constraints in (2). Namely, there is some feasible solution x that incurs a cost of C monitory units, if and only if $\max_{k=1,\ldots,K} \left\{ f^k(\hat{x}) - \alpha_k \right\} \leq 0$. We then search for the minimal cost C for which a feasible solution exists, in a binary search fashion; the above procedure of solving (13) and validating the solution feasibility is used to either return the feasible solution or determine that one does not exist.

Our validation mechanism is again based on simulation. We generate the least number of samples for which the calculated confidence intervals with confidence level α either indicate that there is some constraint that is broken or that all constraints are met with accuracy δ.

We apply the *Saddle Point Mirror Descent SA* that is proposed in Juditsky et al. 2008 to solve (13). Similarly to the Robust SA, in order to guarantee desired accuracy δ with confidence level ε, the step size γ and number of iterations T is calculated based on the problem parameters. We denote by $\text{SaddleSA}(C, \delta, \varepsilon)$ the procedure that solves (13) with cost C, accuracy δ and confidence level ε.

Next, we define the range of the cost in which we perform the binary search.

For this purpose, we calculate an upper bound for the optimal cost. Since we cannot assume anything regarding the routing scheme, we consider the case where each class of customers i is routed to pool j which maximizes the cost of the required staffing level N_{ij} for meeting the SL constraint. We can compute the required staffing levels for typical SL functions using a simple stationary queueing system (such as Erlang-C, or Erlang-A). For other SL functions that cannot be computed in this way, we replace with the constraint $\mathbb{P}\{W > 0\} = 0$ (practically, smaller than some threshold) which is the strongest constraint possible in the sense that satisfying the constraint $\mathbb{P}\{W > 0\} = 0$ implies satisfying any other SL constraint. Let

$$x_{ij} = \begin{cases} N_{ij}, & j \in \arg\max_k c_k N_{ik}; \\ 0, & \text{otherwise.} \end{cases}$$

and let $x_j = \sum_{i=1}^{I} x_{ij}$. Then our upper bound is given by $C_{max} = c^T x$.

Finally, the Constraints Satisfaction Algorithm is displayed in Figure 2.

Theorem 2. *Following the constraint Satisfaction procedure, with accuracy of δ, cost accuracy δ_C and confidence level ε in the sub-procedures Feasible and MirrorSaddleSA, we achieve the following:*

1. *Feasibility:* $\mathbb{P}\left\{ \max_{k=1,\ldots,K} \left(f^k(\hat{x}) - \alpha_k \right) > \delta \right\} \leq 1 - (1-\varepsilon)^{\lceil \log \frac{C_{max}}{\delta_C} \rceil}$
2. *Optimality:* $\mathbb{P}\left\{ c^T \hat{x} - c^* > \delta_C \right\} \leq 1 - (1-2\varepsilon)^{\lceil \log \frac{C_{max}}{\delta_C} \rceil}$

Algorithm: Feasible(x,δ,ε)
Input: solution x, accuracy δ, confidence level ε
Initialization: For each $k=1,\ldots,K$, set the confidence interval $[lb_k,rb_k]=[0,1]$

Step 1 Generate n samples

Step 2 Calculate the confidence interval $[lb_k,rb_k]$ based on all samples

Step 3 If $ub_k \leq \alpha_k+\delta \ \forall k=1,\ldots,K$ return 'True'

Step 4 If $\exists$k s.t. $lb_k \geq \alpha_k+\delta$ return 'False'

Step 5 Go to Step **1**

Algorithm: Constraints Satisfaction$(\delta,\delta_C,\varepsilon)$
Input: feasibility accuracy δ, cost accuracy δ_C, confidence level ε
Initialization: $dC:=C_{max}$, $x^*:=x$;

Step 1 if $dC\leq\delta_C$ return the solution x^*;

Step 2 $dC:=dC/2$;

Step 3 if **Feasible**(x,δ,ε) returns 'True' then set $x^*:=x$, $C:=C-dC$, $x:=x\frac{C}{(C+dC)}$. Go to Step 1;

Step 4 $x:=$ SaddleSA(C,δ,ε);

Step 5 if **Feasible**(x,δ,ε) returns 'True' then Set $x^*:=x$, $C:=C-dC$, $x:=x\frac{C}{(C+dC)}$ Go to Step 1;

Step 6 $C:=C+dC$, $x:=x\frac{C}{(C-dC)}$. Go to Step 1;

Figure 2: Constraint Satisfaction Algorithm

Proof. First, note that the CS algorithm terminates after exactly $\lceil\log\frac{C_{max}}{\delta_C}\rceil$ iterations. Denote by $\hat{x}_{(i)}$ the solution obtained by the i-th iteration. Furthermore, we say that a cost C is feasible if there exists a δ-feasible solution $\hat{x}$, that is $\max_{k=1,\ldots,K} f^k(\hat{x})-\alpha_k<\delta$ and $c^T\hat{x}=C$. Then,

1. The probability that *Feasible* $\left(\hat{x}_{(i)}\right)$ returns the answer 'True' given that $\hat{x}_{(i)}$ is not feasible is bounded by ε. Hence, the probability that the final solution $\hat{x}$ is not feasible is bounded by the probability that in at least one iteration i of the algorithm, the sub-procedure Feasible wrongly declared $\hat{x}_{(i)}$ as feasible. Since there are $\log C_{max}$ iterations, we deduce the result.
2. There are two events by which the algorithm will fail to identify a feasible solution, even though such one exists:
   Event A: The sub-procedure SaddleSA fails to find a δ-optimal solution. The probability of this event is bounded by ε
   Event B: SaddleSA successfully finds aa δ-optimal solution which is wrongly denied by the sub-procedure Feaisble. The probability of this even is bounded by $\varepsilon(1-\varepsilon)$
   The probability that at some iteration i with a feasible cost C_i, the algorithm will fail to identify a feasible solution is thus bounded by $2\varepsilon-\varepsilon^2<2\varepsilon$. In the worst case, the cost will be feasible at all iterations and hence the result.

□

4 NUMERICAL EXPERIMENTS

In this section we describe our numerical study. We provide some results from our experiments and describe them in general outlines. For full details and comprehensive analysis, the reader is kindly referred to <Zohar Dissertation>. We have

tested our algorithms on various systems. We have experimented with different kinds of designs including the **V**-model - a model with two types of customers and one type of agents, the **N**-model - a model with two types of customers and two types of agents, one type can serve both types of customers and one is dedicated for one type of customers, and the **M**-model - a model with two types of customers and three types of agents, one type can serve all types of customers and the other two types are dedicated each to a different type of customers. We have also considered several types of routing policies. Specifically, we considered the Static Priority (SP), the Threshold Priority (TP) and the Fixed-Queue-Ratio (FQR). According to SP, each type of customer has a prioritization over the servers types, and each type of server has a prioritization over the types of customers. Arriving customers are routed to an available server with the highest priority; when becoming available, a server will admit the earliest arriving customer of the highest priority type. TP is similar to SP with the additional constraint that a customer of type i can be admitted to service only if the number of idle servers exceeds some threshold τ_i. FQR is used to maintain (pre-defined) fixed ratios between the queues length of all the types of customers, and fixed ratios between the number of idle servers of each type. We have also experimented with different types of service distributions $\mathscr{D}^S$, patience distribution $\mathscr{D}^P$ and various types of service level objectives.

The first part of our study concentrated on stationary models. We used simulation in order to reconstruct the functional objectives and constrains $f^k(N)$ from our problems (1) and (2). We used this for two purposes: first, it allowed us to check the correctness of the assumption about convexity, and second, it was used in order to identify the (approximated) optimal solution and objective which consequently provided a solid ground for validating our results. In some cases, the service level function seemed to be convex. It was then evident that our algorithms converged to the true optimal solution or very close to it. In some other cases, however, not all the service levels were convex on the entire search space. In these cases, we defined additional linear constraints stemming from stabilization consideration, i.e. necessary conditions for the system to stabilize and not explode at the long run. This limited the search to a region on which all the service levels are very likely to be convex. Indeed, we were successful in attaining optimal or close-to-optimal solutions.

Tables 1 and 2 summarize some of our experiments results. In this tables we used $\mu = (\mu_{11},\ldots,\mu_{1I},\mu_{21}\ldots\mu_{JI})$ to denote the vector of service rates, i.e. the inverse of the average service times. μ_{ji} corresponds to the service rate by which agents from type j serve customers from class i. For the Log-Normal distribution, we used a standard deviation $\sigma = \frac{1}{\mu}$, i.e. same coefficient of variation as the exponential distribution; In the same way we used θ to denote the vector of abandonment rate when it was relevant, *SL* denotes the type of service level that appear either in the objective function or in the constraints, depending on the formulation of the problem. In the former, we denote by c_S the vector of cost coefficients of the service levels in objective function, and in the latter we specify the constraints bounds by α. The staffing cost vector is denoted by c_N. At the bottom section, we describe the results: We present the solution obtained by our algorithm $\hat{N}$, the true optimal solution N^* as derived by simulation, and the difference between the objective function value of $\hat{N}$ and N^*, $\hat{f}-f^*$.

Table 1: Cost Optimization Results

	Examples	
	1	2
Model	N	N
$\mathscr{D}^S$	*exp*	*exp*
$\mathscr{D}^P$	-	*exp*
Routing	SP	SP
μ	$(1,0,1.5,2)$	$(1,0,1.5,2)$
θ	-	$(1,1)$
SL	$\mathbb{E}[W]$	$\mathbb{P}\{ab\}$
c_N	$(1,2)$	$(1,2)$
c_S	$(20,10)$	$(300,200)$
$\hat{N}$	$(80,54)$	$(98,58)$
N^*	$(88,47)$	$(102,56)$
$\hat{f}-f^*$	1.97	1.1
$\frac{\hat{f}-f^*}{f^*}$	0.01	0.005

Table 2: Constraints Satisfaction Results

	Examples					
	1	2	3	4	5	6
Model	V	N	N	N	N	M
$\mathscr{D}^S$	*exp*	*exp*	*exp*	*logn*	*exp*	*exp*
$\mathscr{D}^P$	-	-	-	-	*exp*	-
Routing	TP	SP	FQR	SP	SP	SP
μ	$(1.5,2)$	$(1,0,1.5,2)$	$(1,0,1.5,2)$	$(1,0,1.5,2)$	$(1,0,1.5,2)$	$(1,0,1.5,2,0,2.5)$
θ	-	-	-	-	$(1,1)$	-
SL	$\mathbb{P}\{W>0\}$	$\mathbb{P}\{W>0\}$	$\mathbb{P}\{W>t\}$	$\mathbb{P}\{W>0\}$	$\mathbb{P}\{ab\}$	$\mathbb{P}\{W>0\}$
c_N	$(1,1)$	$(1,2)$	$(1,2)$	$(1,2)$	$(1,2)$	$(1,2,2)$
α	$(0.2,0.5)$	$(0.2,0.5)$	$(0.2,0.2)$	$(0.2,0.5)$	$(0.05,0.1)$	$(0.2,0.5)$
$\hat{N}$	172	$(83,68)$	$(91,60)$	$(92,62)$	$(79,63)$	$(105,37,9)$
N^*	172	$(85,67)$	$(92,59)$	$(92,62)$	$(80,60)$	$(105,39,7)$
$\hat{f}-f^*$	0	0	1	0	3	0
$\frac{\hat{f}-f^*}{f^*}$	0	0	0.0046	0	0.0146	0

The probably more interesting cases, practically speaking, are systems in which arrival rates vary with time. Consequently, staffing levels must also be allowed to change with time in order to meet with the timely constraints or align with the timely cost optimization.

In the most general case, we seek to optimize a system on a given time horizon, which may differ from **Service Level interval** (the time basis on which the Service Levels are measured) and from the **planning period** (the shortest amount of time over which staffing must remain constant). For instance, we may wish to optimize the staffing levels during a whole week, satisfying daily service level constraints, and considering hourly planning period. We are dealing with this situation as follows. We simulate sample paths over the entire time horizon and gather the relevant statistics aggregated by the Service Levels intervals. This enables us to differentiate the Service Level with respect to the staffing level in each pool, on each planning period. We will thus have to make the assumption that the service level functions are convex in the $J \times T$-dimensional staffing vector, where J is the number of pools and T is the number of planning periods. To facilitate, following the example from above, in each iteration of our algorithm, we run a simulation over the entire week and gather the relevant statistics aggregated by day. For each daily Service Level, we estimate the gradient by increasing the staffing levels of each pool in each hour of the week and measure the Finite Differences.

It may be also possible to consider decomposing the problem into smaller independent problems. This can be achieved by first dividing the time horizon into intervals of the same length of the Service Level intervals, and then treat the global optimization problem as several smaller independent problems that are solved sequentially, interval by interval. The state at the end of each interval is taken as the state at the beginning of the consecutive interval. This approach could potentially accelerate the computations without compromising much on global optimality. We applied this approach to realistic example. In this example, we took data of a random day from a real medium-size Call Center, providing different types of banking services. We focused on two types of calls: calls coming from Business customers and Quick Request calls. The arrival rates of both types of incoming calls seem very typical to a standard call center (see Figure 3). Furthermore, we fitted a distribution for the service times, assuming that service times are dependent of the type of customer, but do not depend on the server taking the call. The log-normal seemed like the best fit. We also fitted a distribution for the patience of customers, i.e. the time they are willing to wait for service before they abandon the call. In this case, the patience of both customer types fitted to the exponential distribution. There are two types of servers in this system. One type is catering to both customer types, while the other is dedicated only to Business customers, and their cost are similar. Based on this setting, we have compared between two optimization problem. In both problems, we took a planning period of one hour (i.e. we allowed the staffing to change on at each hour of the day). In the first case, we demanded that not more than 10 percents of Business customer and not more than 50 percents of Quick Request customers will have to be delayed on any hour (i.e. hourly service-level interval). In the second case, we relaxed the service level constraint to be measured on daily basis (daily

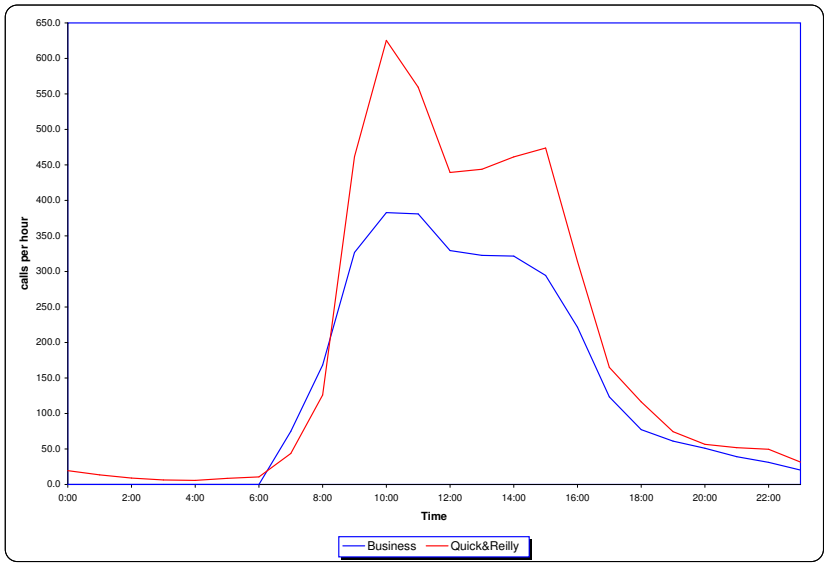

Figure 3: Call Volumes to a Medium-Size Call-Center

service-level interval). Results revealed that in the second case, our algorithm found a solution with a corresponding cost that was 11% lower than the cost obtained in the first case. However, in this case, while the service levels were very good during rush hour (8:00 am until 17:00 pm), essentially all the customers that arrived outside of these hours delayed and encountered a bad service experience (average wait is several minutes). Smaller difference were detected when we tried to satisfy constraints on the probability to abandon both hourly and daily.

5 CONCLUSIONS

In this work we developed simulation-based algorithms for identifying optimal staffing of systems with skills-based-routing, given the routing scheme. An interactive version of our SBR simulator will be installed on <SEElab Server>. These algorithms can be used for very general settings, including time-varying models and general distributions, to either optimize service levels and labor costs or optimize labor costs while satisfying some desirable Service Level constraints.
The algorithms were proven to work very well, and in most cases attained the optimal solution even when the service levels were not convex in the staffing levels. However, in few other cases, the non-convexity nature of the service levels caused the algorithms to converge to sub-optimal solution. Although this can be controlled by applying simple procedures, it is interesting to try to find alternative algorithms that do not rely on convexity, and might even work faster.
While our solution can be very practical for many applications, it lacks the ability to incorporate labor scheduling constraints and thus can not be used to provide an optimal scheduling solution.
One important future direction is to develop algorithms, or combine the ones that are introduced in this work, with a scheduling mechanism which is traditionally carried out by some mathematical programming solver.
Another potential direction is to provide optimal solution for more complex systems with complex processes, such as Service Networks, Petri Nets etc. This will probably require more sophisticated algorithms.

REFERENCES

Armony, M., and A. Mandelbaum. 2004. Design, staffing and control of large service systems: The case of a single customer class and multiple server types. *Draft* 28:271–289.

Atlason, J., E. Epelman, and S. Henderson. 2004. Call-center staffing with simulation and cutting plane methods. *Annals of Operations Research* 127:333–358.

Cezik, T., and P. L'ecuyer. 2008. Staffing mutiskill call centers via linear programming and simulation. *Management Science* 54 (2): 310–323.

Glynn, P. 1987. Limit theorems for the method of replications. *Stochastic Models* 33:343–355.

Gurvich, I., M. Armony, and A. Mandelbaum. 2008. Service level differentiation in call centers with fully flexible servers. *Management Science* 54:279294.

Gurvich, I., and W. Whitt. 2008. Asymptotic optimality of queue-ratio routing for many-server service systems. *Draft* 33:7584.

Gurvich, I., and W. Whitt. 2010. Service level differentiaition in many-server service systems: A solution based on fixed-queue-ratio routing. *Operations Research* 58:316328.

Ho, Y., and X. Cao. 1983. Optimization and perturbation analysis of queueing networks. *Journal of Optimization Theory and Applicationsw* 40:559–582.

Juditsky, A., G. Lan, A. Nemirovski, and A. Shapiro. 2008. Stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* 19:1574–1609.

Polyak, B., and A. Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* 30:838–855.

Robbins, H., and S. Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22:400–407.

Wallace, R., and W. Whitt. 2005. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management* 7:276–294.

AUTHOR BIOGRAPHIES

ZOHAR FELDMAN is a Research Staff Member at IBM Research Labs in Haifa, focusing on the development and application of operation research methods for practical business optimization problems. He is currently a Ph.D. candidate at the Technion. His main research interests include stochastic optimization and control, and simulation-based methods. His email address is zoharf@il.ibm.com.

AVISHAI MANDELBAUM is the Benjamin & Florence Free professor, at the faculty of Industrial Engineering and Management, Technion, Israel. He has been doing research in the area of stochastic processes, from the perspectives of operations research, statistics, engineering, and management. His research has been applied mainly to service operations, with a focus on telephone call centers and emergency departments. His email is avim@tx.technion.ac.il.