A Simulation-based Decomposition Approach for Two-stage Staffing Optimization in Call Centers under Arrival Rate Uncertainty

Thuy Anh Ta * Wyean Chan * Fabian Bastin * Pierre L'Ecuyer *

August 7, 2019

Abstract

We study a staffing problem in multi-skill call centers. The objective is to find a minimal-cost staffing solution while meeting a target level for the quality of service to customers. We consider a situation in which the arrival rates are unobserved random variables for which preliminary forecasts are available in a first stage when making the initial staffing decision. In a second stage, more accurate forecasts are obtained and the staffing may have to be modified at a cost, to meet the constraints. This leads to a challenging two-stage stochastic optimization problem. Given the complexity of the queueing model, the quality of service is estimated by simulation for a large number of scenarios and days. To solve this staffing problem in reasonable time, we propose a simulation-based approach that combines sample average approximation with a decomposition method. We provide numerical illustrations based on three call center examples to show the practical efficiency of our decomposition approach. The proposed method could be adapted to several other staffing problems with uncertain demand, e.g., in retail stores, restaurants, healthcare facilities, and other types of service systems.

1 Introduction

Call centers play a major role in businesses and in public service systems. They are used to provide information and assistance, order food, taxis, or other products or services, receive emergency calls, etc. In multi-skill centers, calls are categorized by the type of service requested. Each call type requires a specific skill and each agent has a subset of all the skills. The agents are partitioned into groups in which all have the same skill set. See Gans et al. (2003) and Koole (2013) for more details.

*Department of Computer Science and Operations Research, Université de Montréal, Canada. Member of GERAD and CIRRELT.

The *quality of service* (QoS) is often measured by the *service level* (SL), defined as the fraction of calls answered within a given time limit, called the *acceptable wait threshold* (AWT). Selecting a *staffing* means choosing how many agents of each skill set to have in the center. Each agent has a cost that depends on its skill set. The staffing problem consists in finding a staffing that minimizes the total cost, under a set of constraints on the QoS. In applications, the day is usually divided into periods of 15 to 60 minutes and a staffing is selected for each period, based on distributional forecasts of arrival rates or call volumes (Ceşik and L'Ecuyer, 2008, Ibrahim et al., 2012). A more difficult problem not considered here is the *scheduling* problem (Avramidis et al., 2010), in which a set of admissible shift schedules is first specified, and the decision variables are the number of agents of each group in each shift.

There are two important issues with most staffing methods proposed in the literature: (i) the arrival rates are often assumed perfectly known, and (ii) the QoS targets (constraints) are usually defined with respect to the long-term expected value, which is an average over an infinite number of days. Perfect knowledge of the arrival rates leads to simpler optimization problems, but arrival rates in real-life call centers are uncertain and depend on multiple factors, such as the day of the week, time of the day, level of busyness, holidays and special events, etc.; see for instance (Channouf et al., 2007, Ibrahim et al., 2016, Oreshkin et al., 2016). The QoS for a given day should then be modeled as a random variable. A manager who wants to meet the QoS targets for a given proportion of the days, or with a given probability, should impose distributional or chance constraints. This is especially true if the distribution of the random variable is unbounded or if the distribution is bounded but the upper bound would lead to far too conservative solutions. In other terms, one cannot satisfy the QoS constraints for all scenarios, or doing so would be much too expensive. This motivates the use of chance constraints. The aim of this paper is to address the aforementioned issues by formulating and solving a staffing problem under arrival-rates uncertainty and probability constraints.

We consider a chance-constrained two-stage staffing problem with recourse for a multi-skill call center. The first-stage consists in selecting a staffing based on an initial forecast of the arrival rates, typically made several days in advance, and with a high level of uncertainty. In the second-stage, a more accurate forecast becomes available, and recourse actions may be applied to correct the initial staffing by adding or removing agents at the price of penalty costs. Chance constraints are imposed on the QoS of the day: The recourse must (always) be chosen so that the QoS meets its target with a minimum probability threshold, given the updated forecast. We solve this problem using a *sample average approximation* (SAA) method. This is challenging, due to the nonlinearity of the chance constraints, the large number of integer variables, and the fact that the QoS can only be estimated by Monte Carlo simulation. Previous studies suggest that the chance constraints can be approximated by linear cuts and that the resulting two-stage linear program can be solved directly by standard mixed-integer program (MIP) solvers such as CPLEX. However, the computation time becomes excessive for large instances.

To address this computing cost issue, we propose a simulation-based decomposition method that consists of two main steps. First, for each scenario, we use simulation to generate linear cuts

to remove infeasible solutions. Then, we iteratively solve the two-stage stochastic programming problem in which the chance constraints are replaced by linear ones and we add more cuts whenever a solution does not satisfy the chance constraints. The first step permits one to create linear outer approximations of the probability functions and to linearize the chance constraints. This step is based on the cutting plane method (Atlason et al., 2004), a popular approach to deal with "S-shaped" constraints. This approach is formally justified only when we are in the concave regions of the probability functions. In our context, these concave regions are difficult to identify accurately, and we propose a heuristic that adjusts the staffing to make it very likely that the cuts are generated from the concave regions. To efficiently solve the resulting two-stage linear programs at the second step, we propose a way to strengthen the linear cuts by mixed-integer rounding inequalities (Nemhauser and Wolsey, 1990) and we decompose the mixed-integer linear problems using the L-shaped method (Birge and Louveaux, 2011). With this method, instead of solving the complete mixed-integer program directly, we decompose it and iteratively solve a master program that is enriched by linear cuts at each iteration.

We report numerical experiments for staffing problems over a single time period. Our objective of this paper is not to solve realistic problems based on real data, but to explore the efficiency of the decomposition approach. We solve problems ranging from a small example with 2 call types and 2 agent groups to an example of moderate size with 15 call types and 20 agent groups. In these examples, our simulation-based decomposition approach returns good staffing solutions significantly faster than the deterministic equivalence approach examined in Chan et al. (2016).

In the remainder, we review the relevant literature on the staffing and scheduling of multi-skill call centers in Section 2. In Section 3, we define the two-stage staffing optimization problem and its SAA formulation. We present our decomposition algorithm in Section 4. In Section 5, we compare the performance of the proposed algorithm and the deterministic equivalent approach in multiple numerical experiments. Conclusions are given in Section 6.

2 Literature Review

Much of the research on call centers has focused traditionally on single-skill centers, with a single call type (Avramidis and L'Ecuyer, 2005, Gans et al., 2003, Green et al., 2003). Multi-skill centers involve routing rules, priorities, etc., and are analytically much more complex than a single queue with a single type of customer. There are no known accurate approximation formulas for QoS measures for them, so these measures must be estimated by computationally-costly simulation. For a multi-skill staffing problem with known arrival rates, Cezik and L'Ecuyer (2008) developed a simulation-based MIP optimization method where linear cuts are added iteratively using estimated subgradients of the SL function. Avramidis et al. (2010) extended this method to solve a shift scheduling problem with multiple periods. These methods are in fact adaptions and generalizations of the method of Atlason et al. (2004) for agent scheduling in single-skill call centers with constraints on the expected SL over an infinite time horizon. The

latter method combines simulation with integer programming and cut generation, based on the concavity property of the SL function in the Erlang C model, when the queue is in steady state. However, the concavity property does not necessarily hold in the multi-skill context, so the methods then become heuristic, but they have been shown to work well empirically. Avramidis et al. (2009), Pot et al. (2008), Wallace and Whitt (2005) proposed other algorithms for the single-period staffing problem that use crude approximation formulas, search methods, and corrections by simulation. Call routing is also an important aspect that interplays with routing and scheduling in multi-skill centers: changing the routing policy often changes the optimal staffing solution, and vice-versa; see Chan et al. (2014).

The exact arrival rates are usually unknown, and several authors consider stochastic optimization to capture this uncertainty. Liao et al. (2012) and Liao et al. (2013) model the uncertain arrival rates by discrete probability distributions. Gurvich et al. (2010), Helber and Henken (2010) and Robbins and Harrison (2010) use random sampling from continuous arrival rate distributions, and Gans et al. (2015) explore the Gaussian quadrature for scenarios generation. Robbins and Harrison (2010) consider a stochastic scheduling problem for a single-skill call center, where a penalty cost is given for missing the SL target. Gans et al. (2015) investigate a two-stage scheduling problem with recourse for single-skill call centers. The forecast is updated during the day, and the schedules can be corrected by adding or removing agents for the latter part of the day. These two papers use a MIP solver to deal with a MIP where a set of constraints are generated beforehand by the linearization of the SL and the abandonment ratio, which are taken as the steady-state values given by the analytic formulas for an M/M/s queue with abandonments. It is unclear if and how this approach can be generalized to the multi-skill case, for which no analytic formula is available.

For multi-skill call centers with random arrival rates, Harrison and Zeevi (2005) and Bassamboo et al. (2006) approximate the level of abandonments by a fluid system, and solve a two-stage scheduling problem. Their models seek to minimize the scheduling cost function with a penalty cost on the abandonment ratio. The first-stage decision variables are the schedules, and the second-stage ones control the work assignment of each agent. A major drawback when optimizing a fluid system is that it assumes implicitly some kind of idealistic fluid routing policy, which is unrealistic. Gurvich et al. (2010) optimize a two-stage staffing problem with chance constraints on the steady-state abandonment ratio, for stochastic arrival rates. The requirement is that the QoS can be violated on at most a fraction δ of the arrival rate realizations, where δ represents the level of risk tolerance. Chan et al. (2016) propose an extension of the cutting plane method presented in Ceşik and L'Ecuyer (2008) to solve a two-stage staffing problem with chance constraints on the SL over the day (not the long term SL). The second-stage decision variables are recourse actions to add or remove agents. The present paper considers a similar model and proposes improvements in the method of solution, based on a decomposition approach.

3 Problem Formulation and Sample Average Approximation

We now formulate the two-stage optimization problem considered in this paper. We also give a SAA formulation in which the constraints are approximated by sampling averages. These formulations are similar to those in Chan et al. (2016) and Ta et al. (2018).

3.1 Call Center Model

Consider a call center with K call types indexed from 1 to K, and I agent groups numbered from 1 to I. Agents in group i have the skill set $\mathcal{S}_i \subseteq \{1, \ldots, K\}$, which is the set of call types they can serve. Conversely, $\mathcal{G}_k = \{i : k \in \mathcal{S}_i\}$ is the set of groups that can handle calls of type k. Let $z = (z_1, \ldots, z_I)^{\mathrm{T}}$ be the staffing vector, which gives the number of agents in each group. We assume that calls of type k arrive from a time-homogeneous Poisson process with unknown rate Λ_k over the entire period, where the Λ_k are independent random variables with bounded support but otherwise arbitrary distributions. This models the forecasting uncertainty.

Agents in the same group are homogeneous and when an agent in group i serves a call of type k, the service time has a known distribution, for each pair (i, k). A call abandons the queue (and the call center) when its waiting time exceeds its patience time, which is a random variable with known distribution that may depend on the call type k. Calls are assigned to agents by an arbitrary routing policy whose details are not important. In this paper, we do not optimize the routing policy; we assume it is fixed. One advantage of simulation-based optimization is that there is no need to impose a specific form of routing policy or specific family of probability distributions in the model. For example, the service time can be exponential, lognormal, or gamma, etc. As in Wallace and Whitt (2005), Cezik and L'Ecuyer (2008), and many others, we optimize the staffing for a single time period.

3.2 Service Level Constraints

We measure the QoS by the SL introduced in Section 1, defined as the proportion of callers who wait less than an *acceptable waiting time* (AWT) parameter τ over a given finite time period. This SL is a random variable and the constraints will be probabilistic: the SL must reach a certain target $l \in [0, 1]$ with probability at least $1 - \delta$ for a given $\delta > 0$. The SL can actually be defined in various ways, depending on how we count abandons, the calls that overlap two or more periods, etc. Here we use a popular definition, implemented (among others) in ContactCenters (Buist and L'Ecuyer, 2005). For a staffing vector z and AWT threshold τ, we define

$$\mathrm{SL} = \mathcal{S}(z) = \frac{A(z)}{T - L(z)} \tag{1}$$

where T is the total number of calls that arrived in the period, $A(z)$ is the number of those calls served after waiting at most τ, and $L(z)$ is the number of them that abandoned after

waiting more than τ. For other definitions, see Jouini et al. (2013). Several authors replace T, A, and L by their (transient or steady-state) expectations; see for example Atlason et al. (2004), Avramidis et al. (2009, 2010), Cezik and L'Ecuyer (2008). Then the SL is a constant, defined as a ratio of expectations, instead of a random variable, and the constraints are no longer probabilistic.

We define our chance constraints as follows. For each call type k, we select an AWT τ_k and denote $S_k(z)$ the SL for call type k during the selected period, given the staffing vector z. Let $S_0(z)$ denote the aggregate SL for all calls, with AWT τ_0, over the period. The random variables $S_0(z), \ldots, S_K(z)$ have distributions that depend on z. The constraints are:

$$\mathbb{P}[\mathcal{S}_k(z) \geq l_k] \geq 1 - \delta_k, \qquad k = 0, 1, \ldots, K,$$

where the l_k are SL targets and the $\delta_k \in (0, 1)$ are given risk thresholds.

3.3 Staffing Problem with Recourse

We now describe the two-stage staffing problem. In the first stage, based on an initial forecast that provides a prior distributions for the random arrival rate Λ_k for each call type k, the manager must select an initial staffing $x = (x_1, \ldots, x_I)^{\mathrm{T}}$ at the corresponding cost per agent of $c = (c_1, \ldots, c_I)^{\mathrm{T}}$. Stage 1 can be days or weeks in advance of the target date. Close to the target period (e.g., the previous day or a few hours before), additional information becomes available that can improve the forecast of the arrival rates Λ_k. Let $\xi \in \Xi$ denote this new information. It could be related to weather conditions, the observed number of arrivals in the preceding period, etc. Let $\mathbb{E}_\xi$ denotes the expectation with respect to ξ and $\mathbb{P}[\cdot \mid \xi]$ be the probability distribution conditional on ξ. In particular, the distribution of Λ_k conditional on ξ is not the same as the unconditional one. It usually has a smaller variance.

In the second stage, the manager observes the realization of ξ and based on that, the initial staffing can be modified by adding or removing agents at some penalty costs (by calling them to work at the last minute or offering them to go back home, or canceling meetings, etc.). Note that even in the extreme case where the Λ_k are known exactly conditional on ξ, there is still uncertainty in the second stage and the SL is still a random variable. The recourse in Stage 2 consists in modifying the initial staffing by adding $r_i^+(\xi)$ extra agents to group i at a cost of $c_i^+ > c_i$ per agent, or removing $r_i^-(\xi) \leq x_i$ agents in group i to save c_i^- per agent, where $0 \leq c_i^- < c_i$. After the recourse, the new number of agents in group i is $z_i(\xi) = x_i + r_i^+(\xi) - r_i^-(\xi)$. Let c, c^+, c^-, and $z(\xi)$ be the vectors with components c_i, c_i^+, c_i^-, and $z_i(\xi)$, respectively. We define the recourse vectors as $r^+(\xi) = (r_1^+(\xi), \ldots, r_I^+(\xi))^{\mathrm{T}}$, and $r^-(\xi) = (r_1^-(\xi), \ldots, r_I^-\xi))^{\mathrm{T}}$. Given a staffing $z(\xi)$, the SL for call type k and the aggregate SL are random variables $\mathcal{S}_k(z(\xi))$ for $k = 1, \ldots, K$, and $\mathcal{S}_0(z(\xi))$, respectively. Let $g_k(z; \xi) = \mathbb{P}[\mathcal{S}_k(z) \geq l_k \mid \xi]$ for $k = 0, \ldots, K$.

With this, we have the following chance-constrained staffing problem with recourse:

$$(\mathbf{P1}) \quad \begin{cases} \min\limits_{x \in X} & c^{\mathrm{T}}x + \mathbb{E}_{\xi}\left[Q(x,\xi)\right], \\ \text{where} & Q(x,\xi) = \min & \left\{(c^{+})^{\mathrm{T}}r^{+}(\xi) - (c^{-})^{\mathrm{T}}r^{-}(\xi)\right\} \\ & \quad \text{subject to} & x + r^{+}(\xi) - r^{-}(\xi) = z(\xi), \\ & & g_k(z;\xi) \geq 1 - \delta_k, \quad k = 0, \ldots, K, \\ & & r^{+}(\xi), r^{-}(\xi) \geq 0 \text{ and integer}, \end{cases}$$

in which $X \subset \mathbb{N}^I$ is the support set of first-stage solutions. The constraints in this formulation are on the SL, but they could also be on other QoS measures such as average waiting times, abandonment ratios, etc. We emphasize that in this formulation, for *any realization* of ξ, the recourse must be selected so that the probabilistic constraints are satisfied. Without this assumption, one could be tempted to put no staffing at all on certain days in which ξ takes a bad value, to save costs. Under our model, this is not allowed.

3.4 The SAA Formulation

Instead of solving the two-stage problem (**P1**), we will solve a SAA version. We generate N *scenarios* (realizations of ξ) by Monte Carlo. Let ξ_n denote the realization of ξ under scenario n. Each scenario provides a different distribution of the Λ_k's, conditional on ξ_n, for the second stage. Then we define a discrete probability distribution over these N scenarios by giving probability $p_n > 0$ to scenario n, where $\sum_{n=1}^{N} p_n = 1$. In our numerical examples, we will simply put $p_n = 1/N$ for all n. For scenario n, we denote $r_n^{+} = r^{+}(\xi_n)$, $r_n^{-} = r^{-}(\xi_n)$, and $z_n = (z_{1,n}, \ldots, z_{I,n})^{\mathrm{T}}$. For each ξ_n, we estimate the probability $g_k(z;\xi_n) = \mathbb{P}[\mathcal{S}_k(z(\xi_n)) \geq l_k \mid \xi_n]$ in the constraints of (**P1**) by simulating the call center M times independently, over the given period, conditional on ξ_n. These simulations are also independent across the scenarios. We compute the empirical SL $\hat{\mathcal{S}}_k^m(z_n;\xi_n)$ for each k and each replication m, and we estimate $g_k(z;\xi_n)$ by the proportion of the M replications for which the SL constraint was met:

$$\hat{g}_{k,M}(z_n;\xi_n) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}[\hat{\mathcal{S}}_k^m(z_n;\xi_n) \geq l_k] \quad \text{for } k = 0, \ldots, K,$$

where $\mathbb{I}[\cdot]$ is the indicator function. With these ingredients, the SAA can be written as

$$(\mathbf{P2}) \quad \begin{cases} \min\limits_{x,\, r_n^{+},\, r_n^{-}} & c^{\mathrm{T}}x + \sum\limits_{n=1}^{N} p_n \left[(c^{+})^{\mathrm{T}}r_n^{+} - (c^{-})^{\mathrm{T}}r_n^{-}\right], \\ \text{subject to} & \begin{cases} x + r_n^{+} - r_n^{-} = z_n, \quad \text{for } n = 1, \ldots, N, \\ \hat{g}_{k,M}(z_n;\xi_n) \geq 1 - \delta_k, \quad \text{for } k = 0, \ldots, K,\ n = 1, \ldots, N, \\ x \in X,\ r_n^{+}, r_n^{-} \geq 0 \text{ and integer, for } \ n = 1, \ldots, N. \end{cases} \end{cases}$$

Ta et al. (2018) investigate the convergence properties of this SAA problem. Under reasonable assumptions which hold in call center examples, they show that the optimal value and the solutions of the SAA problem converge almost surely to corresponding ones for the true problem (**P2**) when N and M increase to infinity.

Two important difficulties arise when solving the SAA (**P2**): (i) the constraints $\hat{g}_M(z_n;\xi_n) \geq 0$ are nonlinear and (ii) (**P2**) is expensive to solve when N is large. Chan et al. (2016) handle issue (i) by using a cutting plane method in which the nonlinear constraints are replaced by several linear cuts. This approach can work reasonably well in the simpler situation where there is no recourse but in the two-stage setting, the SAA (**P2**) becomes much more expensive to solve when there is a large number of scenarios. This motivates our introduction of a decomposition method for (**P2**) in the next section.

4 General Methodology with a Decomposition Approach

We now introduce our proposed decomposition approach to solve the two-stage staffing optimization problem. To deal with the chance constraints in the SAA, we use the cutting plane method (Ceşik and L'Ecuyer, 2008) to create outer linear approximations of the (nonlinear) probability functions. This yields two-stage stochastic integer linear programs that could be expensive to solve. We then propose a way to strengthen the linear cuts generated by the cutting plane method, and a simulation-based decomposition algorithm that allows to efficiently find good staffing solutions.

4.1 Cut Generation

We recall the cutting-plane method used by Ceşik and L'Ecuyer (2008) and Atlason et al. (2008) to approximate the chance constraints by linear ones. The method relies on the hypothesis that the SL function is concave in z, at least around the optimal solution. In this paper, instead of being on the expected SL as in those previous papers, our constraints are on a tail probability of the SL, $\mathbb{P}[\mathcal{S}(z) \geq l]$, and it is this probability function rather than the expected SL that is assumed to be concave near the optimum. Chan et al. (2016) observed that these probability functions typically have an S shape: they are convex for small z and concave for large enough z, just like the expected SL.

In our approach, we consider each scenario separately, and for each staffing solution that violates a chance constraint for that scenario, we generate linear cuts based on an (tentative) estimation of the sub-gradient at that staffing point. After adding enough cuts, we obtain a feasible staffing solution for the chance constraints. The result of this procedure is a set of linear cuts that serve as an approximation of the chance constraints and which are used to solve the two-stage problem.

The cutting-plane method is an iterative algorithm that starts at an infeasible solution z and adds new linear cuts based on the sub-gradient of $\hat{g}_{k,M}(z)$ until a feasible solution is obtained.

To avoid starting the algorithm at a null solution ($z = 0$) or in a non-concave region, we use a heuristic that uses a fluid model approximation and adds linear constraints to impose that the system has enough capacity (in the fluid model) to serve at least a fraction α_k of the arriving rate for each call type k. This follows Chan (2013) and Chan et al. (2016). The constraints can be written as

$$\begin{aligned}
\sum_{i \in \mathcal{G}_k} \mu_{k,i} w_{k,i,n} &\geq \alpha_k \Lambda_{k,n}, && k = 1, \ldots, K \\
\sum_{k \in \mathcal{S}_i} w_{k,i,n} &\leq z_i, && i = 1, \ldots, I \\
w_{k,i,n} &\geq 0, && k = 1, \ldots, K,\ i = 1, \ldots, I,
\end{aligned}$$

in which the time unit is the length of the period, $\Lambda_{k,n}$ is the arrival rate of call type k in scenario n, $1/\mu_{k,i}$ is the expected service time for call type k by agent group i, $w_{k,i,n} \geq 0$ represents the (fractional) number of agents of group i assigned to calls of type k in scenario n, and we want to select the parameters α_k so that the initial solution is in a concave region of $\hat{g}_M$. If $\alpha_k < 1$, over an infinite time horizon it would mean that at least a fraction $1 - \alpha_k$ of the calls abandon, on average. Over a finite time horizon, this fraction could be a little smaller because a few of these calls can be served after the end of the period. So if we expect few abandons, it makes sense to initially try α_k close to 1, then iteratively use simulation to estimate the probability values in the constraints, and increase α_k if they are too small. To do this, we select a threshold $\rho > 0$ (e.g., $\rho = 0.5$), and we add agents to the groups that serve call type k if the estimated probability is smaller than ρ. We stop this procedure when all the probability values are larger than ρ. We expect (and assume) that after this, the staffing belongs to the concave region and the sub-gradient cuts are valid.

Then we generate sub-gradient-based linear cuts independently for each scenario, as follows. For scenario n with realization ξ_n, let $g_k^n(z) = \hat{g}_{k,M}(z, \xi_n)$ and let $q_{nk}(z^*)$ denote the estimated sub-gradient of g_k^n at z^*, which is a I-dimensional vector whose element i is defined as

$$q_{nk}^i(z^*) = [g_k^n(z^* + de_i) - g_k^n(z^*)]/d,$$

where e_i is the ith unit vector (with 1 at position i and 0 elsewhere), $d \geq 1$ is an integer, and the simulations required to compute g_k^n at the two different values to obtain the finite difference are always made with well-synchronized common random numbers, as explained in L'Ecuyer and Buist (2006) and Cezik and L'Ecuyer (2008). If the empirical function $g_k^n(z)$ was guaranteed to be always convex in each coordinate, we could always take $d = 1$, but this function is somewhat noisy, so it may fail to be convex even where its expectation is convex, especially when M is small, and it is sometimes safer to take $d = 2$ or 3 for this reason. This issue is discussed in more details in Cezik and L'Ecuyer (2008).

If $q_{nk}(z^*)$ is a sub-gradient of g_k^n at z^*, then we have $g_k^n(z^*) + q_{nk}(z^*)(z - z^*) \geq g_k^n(z)$. Since

we want $g_k^n(z) \geq 1 - \delta_k$, the following inequality must hold:

$$q_{nk}(z^*)z \geq 1 - \delta_k - g_k^n(z^*) + q_{nk}(z^*)z^*. \tag{2}$$

We add this inequality as a constraint (a linear cut) to the second-stage linear program for scenario n, which reads as:

$$\min_{(z,w)\in\mathbb{N}^I\times\mathbb{R}_+^{K\times I}} \{c^{\mathrm{T}}z \mid \ A^n z \leq b^n,\ \mathcal{H}^n z + \mathcal{K}^n w \leq h^n\}, \tag{3}$$

where $A^n z \leq b^n$ refers to the set of sub-gradient cuts and $\mathcal{H}^n z + \mathcal{K}^n w \leq h^n$ are constraints given by the fluid model. The cutting-plane procedure permits one to approximate (**P2**) by a mixed-integer linear programming (MIP) model. Proposition 1 below states that by adding enough cuts to approximate the chance constraints, we can obtain an optimal solution to (**P2**) by solving the corresponding MIP.

Let $\widehat{Q}(x)$ Let $\widehat{Q}(x) = \frac{1}{N}\sum_{n=1}^N \widehat{Q}_M(x;\xi_n)$ denote the value of the second stage of (**P2**) for a given x, where

$$\begin{aligned}
\widehat{Q}_M(x;\xi_n) = \ \min \quad & (c^+)^{\mathrm{T}}r^+ + (c^-)^{\mathrm{T}}r^- & \\
\text{subject to} \quad & \hat{g}_{k,M}(x + r^+ - r^-,\ \xi_n) \geq 1 - \delta_k & k = 0,\ldots,K \\
& r^+, r^- \in \mathbb{N}^I. &
\end{aligned}$$

For each scenario ξ_n, we denote by $\overline{Q}_M(x;\xi_n)$ the value of the second stage after replacing the constraints $\hat{g}_M(z;\xi_n) \geq 0$ by the linear cuts, i.e.,

$$\text{(P3)} \qquad \begin{cases}
\overline{Q}_M(x;\xi_n) = \ \min & (c^+)^{\mathrm{T}}r^+ + (c^-)^{\mathrm{T}}r^- \\
\text{subject to} & A^n(x + r^+ - r^-) \leq b^n \\
& \mathcal{H}^n(x + r^+ - r^-) + \mathcal{K}^n w \leq h^n \\
& r^+, r^- \in \mathbb{N}^I
\end{cases}$$

where $A^n(x + r^+ - r^-) \leq b^n$ are the linear cuts added for scenario n. By incorporating these replacements in (**P2**), we obtain the following large MIP, whose solution approximates the solution of (**P2**):

$$\text{(P4)} \qquad \min_{x\in\mathbb{N}^I} \left\{ \bar{f}(x) = c^{\mathrm{T}}x + \frac{1}{N}\sum_{n=1}^N \overline{Q}_M(x;\xi_n) \right\}.$$

The next proposition says that if the linear cuts are always upper bounds on the chance constraints and we add enough of them, we obtain an optimal solution for (**P2**) by solving (**P4**).

Proposition 1 *Suppose that for each linear cut of the form (2) added to (**P4**), z^* is in the concave region of the probability function g_k^n, and $q_{nk}(z^*)$ is a sub-gradient of this probability function. If $(x^*, \bar{f}^*)$ is an optimal solution and the optimal value of (**P4**) and if (r_n^{*+}, r_n^{*-}) is an optimal solution to (**P3**) such that $\hat{g}_{k,M}(x^* + r_n^{*+} - r_n^{*-}\ ;\ \xi_n) \geq 1 - \delta_k$ for all n and k, then*

$(x^*, \bar{f}^*)$ *is also an optimal solution and the optimal value to (**P2**).*

Proof. Under the given assumption, given a first-stage solution x, we always have

$$\left\{(r^+, r^-) \,\middle|\, \begin{array}{l} \hat{g}_M(x + r^+ - r^-; \xi_n) \geq 0 \\ r^+, r^- \in \mathbb{N}^I \end{array}\right\} \subseteq \left\{(r^+, r^-) \,\middle|\, \begin{array}{l} A^n(x + r^+ - r^-) \leq b^n \\ \mathcal{H}^n(x + r^+ - r^-) + \mathcal{K}^n w \leq h^n \\ r^+, r^- \in \mathbb{N}^I,\ w \geq 0 \end{array}\right\}. \tag{4}$$

We denote by $\{x_1^*, r_{1n}^{*+}, r_{1n}^{*-},\ n = 1, \ldots, N\}$ and optimal solution to (**P2**) and by $\{x_2^*, r_{2n}^{*+}, r_{2n}^{*-},\ n = 1, \ldots, N\}$ and optimal solution to (**P4**). According to (4) we have

$$c^{\mathrm{T}} x_1^* + \frac{1}{N}\sum_{n=1}^{N} (c^+)^{\mathrm{T}} r_{1n}^{*+} - (c^-)^{\mathrm{T}} r_{1n}^{*-} \geq c^{\mathrm{T}} x_2^* + \frac{1}{N}\sum_{n=1}^{N} (c^+)^{\mathrm{T}} r_{2n}^{*+} - (c^-)^{\mathrm{T}} r_{2n}^{*-}. \tag{5}$$

Moreover, if $\hat{g}_{k,M}(x_2^* + r_{2n}^{*+} - r_{2n}^{*-}; \xi_n) \geq 1 - \delta_k$ for all n, k, then $\{x_2^*, r_{2n}^{*+}, r_{2n}^{*+},\ n = 1, \ldots, N\}$ is also a feasible solution to (**P2**), so

$$c^{\mathrm{T}} x_1^* + \frac{1}{N}\sum_{n=1}^{N} (c^+)^{\mathrm{T}} r_{1n}^{*+} - (c^-)^{\mathrm{T}} r_{1n}^{*-} \leq c^{\mathrm{T}} x_2^* + \frac{1}{N}\sum_{n=1}^{N} (c^+)^{\mathrm{T}} r_{2n}^{*+} - (c^-)^{\mathrm{T}} r_{2n}^{*-}. \tag{6}$$

From (5) and (6) we can deduce that $\{x_2^*, r_{2n}^{*+}, r_{2n}^{*+},\ n = 1, \ldots, N\}$ is also an optimal solution to (**P2**). This completes the proof. ■

So in principle, we can obtain an optimal solution to the SAA problem (**P2**) by adding enough linear cuts to the second-stage problems and then solve the MIP (**P4**) via a standard solver such as CPLEX. However, in a large scale setting and as the number of scenarios increases, (**P4**) would be too large and too hard to solve directly. We can then rely on the L-shaped algorithms presented next.

4.2 L-shaped Algorithms

For any first-stage solution x of (**P4**), evaluating this solution requires solving N second-stage sub-problems of the form (**P3**). This suggests an L-shaped decomposition approach for the two-stage problem. However, the problem involves integer variables at both first and second stages, and therefore solving it exactly, even with a decomposition method, is very challenging when N and M are large. The absence of general efficient methods for this type of problem reflects this difficulty (see Birge and Louveaux, 2011, Chapter 7). Several techniques have been proposed over the years, but these techniques are either expensive, or developed under specific restrictions on the two-stage problem, e.g., that the recourse matrix has only integer coefficients, which is not the case in our context. In what follows, we present a simple integer L-shaped algorithm that can be combined with mixed-integer rounding inequalities (in Section 4.3) to efficiently find good integer solutions for the two-stage problem.

The general idea of the L-shaped method is to replace the recourse function (or the second-stage objective function) by a piece-wise linear and convex function. Since the nonlinear objective function at the first stage involves a solution to all the second-stage programs, we want to avoid numerous function evaluations for it. For this, we define a master linear model in x and we only evaluate the recourse function as a sub-problem. We do this by considering a continuous relaxation of the second-stage problem and using the duality properties of this relaxation.

For any first-stage solution x, to get a feasible solution for the second stage, we just need to add a large enough vector r^+ of agents and set $r^- = 0$. This means that the problem (**P1**) has a relatively complete recourse, i.e., that the second-stage problems always have feasible solutions, for any given first-stage solution x (see Birge and Louveaux, 2011, Page 113). In addition, under the concavity assumption, the linear cuts generated from the cutting plane method (Section 4.1) are upper bounds on the chance constraints. This means that the problem (**P1**) has relatively complete recourse, i.e., the second-stage problems always have feasible solutions given any first-stage solution x (see Birge and Louveaux, 2011, Page 113). In addition, under the concavity assumption, the linear cuts generated from the cutting plane method (Section 4.1) are upper bounds on the chance constraints. In other terms, the linearized second-stage problem will be a relaxation of the true second-stage problem, and consequently, any feasible solution of the true second-stage problem will be feasible for the relaxed second-stage problem. Therefore, (**P4**) is also relatively complete. Thus, when applying a L-shaped method to solve (**P4**), we only need to add optimality cuts, i.e., linear cuts to build a piecewise linear function that approximates the recourse function, to the master problem.

We can write the master problem of (**P4**) as

$$(\mathbf{MP1}) \qquad \begin{cases} \min\limits_{x,\theta} & c^{\mathrm{T}}x + \theta \\ \text{subject to} & \Pi x - \mathbf{1}\theta \leq \pi_0 \\ & x \in X \end{cases} \tag{7}$$

where the variable $\theta \in \mathbb{R}$ serves as an underestimation of the second-stage objective function, while the constraints (7) are optimality cuts obtained by relaxing the second-stage problem and generating cuts based on its duality. Suppose the constraints of the second-stage problem for scenario n can be written as $T^n x + W^n y = r^n$, where y is the vector of second-stage variables. In our context, y contains r^+, r^-, and w (coming from the fluid model). For each solution x^* and each scenario $n = 1, \ldots, N$, we rewrite the relaxation of the second-stage problem using equality constraints as

$$\min_y \left\{ q^{\mathrm{T}} y \;\middle|\; T^n x^* + W^n y = r^n,\ y \geq 0 \right\}.$$

We then solve the dual to obtain a dual optimal solution

$$\sigma_n = \arg\max_{\sigma} \left\{ (r^n - T^n x^*)^{\mathrm{T}} \sigma \;\middle|\; (W^n)^{\mathrm{T}} \sigma \leq q \right\}.$$

The duality properties imply that

$$\begin{aligned}
\overline{Q}_M(x;\xi_n) &= \min_y \left\{ q^{\mathrm{T}} y \;\middle|\; T^n x + W^n y = r^n,\ y = (r^+, r^-, w) \geq 0,\ r^+ \text{ and } r^- \text{ integer} \right\} \\
&\geq \min_y \left\{ q^{\mathrm{T}} y \;\middle|\; T^n x + W^n y = r^n,\ y \geq 0 \right\} \\
&= \max_\sigma \left\{ (r^n - T^n x)^{\mathrm{T}} \sigma \;\middle|\; (W^n)^{\mathrm{T}} \sigma \leq q \right\} \\
&\geq \sigma_n^{\mathrm{T}} (r^n - T^n x).
\end{aligned}$$

Since we want $\theta \geq \frac{1}{N} \sum_{n=1}^{N} \overline{Q}_M(x;\xi_n)$, we can add the following optimality cut to the master problem:

$$\theta \geq \frac{1}{N} \sum_{n=1}^{N} \sigma_n^{\mathrm{T}} (r^n - T^n x),$$

or equivalently

$$-\left(\frac{1}{N} \sum_{n=1}^{N} \sigma_n^{\mathrm{T}} T^n \right) x - \theta \leq -\frac{1}{N} \sum_{n=1}^{N} \sigma_n^{\mathrm{T}} r^n. \tag{8}$$

It is also possible to add several cuts per master iteration, based on the idea of the multi-cut L-shaped method (Birge and Louveaux, 2011, Page 198). More precisely, we can partition the set of all scenarios into L disjoint subsets $N_1, \ldots, N_L$ and reformulate (**MP1**) as

$$(\textbf{MP2}) \qquad \begin{cases} \min\limits_{x,\theta_1,\ldots,\theta_L} & c^{\mathrm{T}} x + \sum\limits_{l=1}^{L} \theta_l \\ \text{subject to} & \Pi^l x - \mathbf{1}\theta_l \leq \pi_0^l, \qquad l = 1, \ldots, L \\ & x \in X, \end{cases} \tag{9}$$

where the constraints (9) are optimality cuts given by L subsets of scenarios. For each subset N_l, the following optimality cut can be added to the master problem:

$$-\frac{1}{N_l} \left(\sum_{n \in N_l} \sigma_n^{\mathrm{T}} T^n \right) x - \theta_l \leq -\frac{1}{N_l} \sum_{n \in N_l} \sigma_n^{\mathrm{T}} r^n. \tag{10}$$

We summarize this L-shaped approach in Algorithm 1. If $L = 1$, we have a single-cut L-shaped algorithm in which only one cut is generated per iteration. If $L = N$, we generate cuts for each scenario.

4.3 Strengthening the Cutting Planes

In this section we present a way to strengthen the sub-gradient cuts defined in (2) by using mixed-integer rounding (MIR) inequalities. This approach plays a central role in the development of strong cutting planes for mixed-integer programming. MIR inequalities can be derived from a single mixed-integer constraint, and have been shown to be able to generate all facets inducing valid inequalities for any mixed 0-1 integer program (Nemhauser and Wolsey, 1990).

Algorithm 1: L-shaped algorithm

repeat

Select L clusters of scenarios that form a partition of all scenarios

Solve (MP2) to obtain a solution $(x^*, \theta_1^*, \ldots, \theta_L^*)$

Compute

$$\overline{Q}(x^*) = \sum_{n=1}^{N} \min_{y} \left\{ q^{\mathrm{T}} y \;\middle|\; T^n x^* + W^n y = r^n,\ y \geq 0 \right\}$$

if $\sum_{l=1}^{L} \theta_l^* < \overline{Q}(x^*)$ **then**

Add L optimality cuts to (MP2)

until $\sum_{l=1}^{L} \theta_l^* \geq \overline{Q}(x^*)$;

Return x^* as a first-stage solution

These MIR inequalities can improve the L-shaped algorithm described in the previous section, since this L-shaped method relies on second-stage continuous relaxations.

Consider a sub-gradient cut of the form $\sum_{i=1}^{I} a_i z_i \geq b$. Since the sub-gradients are always generated to be non-negative, we have $a_i \geq 0$ for all $i = 1, \ldots, I$. Let $\mathcal{P} = \{z \in \mathbb{N}^I \mid \sum_{i=1}^{I} a_i z_i \geq b\}$ be the set of feasible solutions under the sub-gradient cuts.

Proposition 2 *The following inequalities hold for all $z \in \mathcal{P}$:*

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} a_t z_t + d_i a_i z_i \geq \left\lceil \frac{b}{a_i} \right\rceil d_i a_i, \qquad \forall\ i = 1, \ldots, I, \tag{11}$$

where $d_i = b/a_i - \lceil b/a_i \rceil + 1$.

Proof. Given $i \in \{1, \ldots, I\}$ such that $a_i > 0$, we can write the inequality $\sum_{i=1}^{I} a_i z_i \geq b$ as

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} \frac{a_t z_t}{a_i} + z_i \geq \frac{b}{a_i},$$

which can be written as

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} \frac{a_t z_t}{a_i} \geq \frac{b}{a_i} + 1 - \left\lceil \frac{b}{a_i} \right\rceil + \left\lceil \frac{b}{a_i} \right\rceil - z_i - 1. \tag{12}$$

Since $z_i \in \mathbb{N}$, we consider the two cases $z_i \geq \lceil b/a_i \rceil$ and $z_i \leq \lceil b/a_i \rceil - 1$. If $z_i \geq \lceil b/a_i \rceil$ then

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} \frac{a_t z_t}{a_i} \geq \left(1 + \frac{b}{a_i} - \left\lceil \frac{b}{a_i} \right\rceil\right) \left(\left\lceil \frac{b}{a_i} \right\rceil - z_i\right) = d_i \left(\left\lceil \frac{b}{a_i} \right\rceil - z_i\right), \tag{13}$$

as the left side of the inequality is non-negative and the right side is non-positive. Moreover, if

$z_i \leq \lceil b/a_i \rceil - 1$, given that $b/a_i - \lceil b/a_i \rceil + 1 \leq 1$, from (12) we obtain

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} \frac{a_t z_t}{a_i} \geq d_i + d_i \left(\left\lceil \frac{b}{a_i} \right\rceil - z_i - 1 \right) = d_i \left(\left\lceil \frac{b}{a_i} \right\rceil - z_i \right). \tag{14}$$

We obtain (11) by combining (13) and (14). ■

We now consider a set of feasible staffing solutions at scenario n after adding sub-gradient cuts and fluid constraints $\mathcal{P}^n = \left\{ A^n z \leq b^n,\ \mathcal{H}^n z + \mathcal{K}^n w \leq h^n \right\}$. Let J be the number of rows of matrix A^n, a^n_{ij} its element on row i and column j, and b^n_j the j^{th} element of vector b^n. The constraints given by the sub-gradient cuts can be strengthened using Proposition 2 as follows. We will use these stronger inequalities in our method.

Corollary 1 *The following inequalities hold for all $z \in \mathcal{P}^n$*

$$\sum_{\substack{t=1,\ldots,I \\ t \neq i}} a^n_{jt} z_t + d^n_i a^n_{ji} z_i \leq \left\lceil \frac{b^n_j}{a^n_{ji}} \right\rceil d^n_i a^n_{ji}, \qquad i \in \{1,\ldots,I\}, \quad j \in \{1,\ldots,J\}, \quad a^n_{ji} \neq 0, \tag{15}$$

where $d^n_{ji} = b^n_j / a^n_{ji} - \lceil b^n_j / a^n_{ji} \rceil + 1$.

4.4 The Simulation-based Decomposition Algorithm

Algorithm 2 summarizes our complete simulation-based decomposition method. The algorithm has two main parts. In the first part, we solve the staffing optimization problem for each scenario separately to approximate the chance constraints by linear cuts. In the second part, we iteratively solve the two-stage stochastic linear programs in which the chance constraints are replaced by linear cuts using the L-shaped approach. If the second-stage solution given by the L-shaped method is found to be unfeasible for the chance constraints, we use simulation to generate more linear cuts (2) to better approximate the chance constraints. This iterative procedure stops when we find first-stage and second-stage solutions that satisfy all the chance constraints. Proposition 3 states that under reasonable conditions, the procedure will stop after a finite number of steps.

Given that the arrival rates are assumed to be bounded, we can always choose a staffing large enough such that all the probability constraints are satisfied. So, without loss of generality we can assume that the set of feasible staffing solutions at the first stage is finite. Steps 1 and 2 of Algorithm 2 are basically a procedure to separately solve the staffing optimization problem for each scenario, i.e., we iteratively generate cuts and solve the corresponding linear programs until getting a staffing solution satisfying all the chance constraints. An important step of the algorithm is that when there is a call type for which the corresponding probability value is too small, then we need to adjust the staffing, as the current staffing does not belong to the concave region of the probability function and would result in bad cuts. Moreover, since the

Algorithm 2: Simulation-based decomposition algorithm with strengthened cuts

 # 1. Initialization

– Select a threshold $\rho > 0$ to determine a "concave region" for the functions $\hat{g}_M$, e.g., $\rho = 0.5$

– Add preliminary constraints using the fluid model approximation

– Select a step size $d \in \mathbb{N}^*$ for the subgradient estimations and $s \in \mathbb{N}^*$

 # 2. Iteratively adding linear cuts

For each scenario $n = 1, \ldots, N$

repeat

  Solve $\min_{z,w} \{c^{\mathrm{T}} z \mid A^n z \leq b^n, \mathcal{H}^n z + \mathcal{K}^n w \leq h^n\}$ to obtain a solution z^*

  # 2.1 For each k with too small prob. value, add s agents to a group that can serve call type k

  repeat

    Run the simulation with staffing z^* to obtain $\hat{g}_M(z^*; \xi_n)$

    $\bar{k} = \operatorname{argmin}_k \hat{g}_{k,M}(z^*; \xi_n)$

    if $\hat{g}_{\bar{k},M}(z^*; \xi_n) < \rho$ **then**

      Select i randomly and uniformly in $\mathcal{G}_{\bar{k}}$ and set $z_i^* = z_i^* + s$

  until $\hat{g}_{\bar{k},M}(z^*; \xi_n) \geq \rho$ *for all* k;

  # 2.2 Add sub-gradient cuts

  for $k = 0, \ldots, K$ **do**

    if $\hat{g}_{k,M}(z^*; \xi_n) < 1 - \delta_k$ **then**

      Add sub-gradient cut (2) to the set $\{A^n z \leq b^n\}$

until $\hat{g}_{k,M}(z^*; \xi_n) \geq 1 - \delta_k$ *for all* k;

– Add valid inequalities for each sub-gradient cuts initialized (as per Corollary 1)

 # 3. Iteratively solving the linear problem and adding more linear cuts

repeat

  # 3.1. Solve the sub-problem to obtain a first- and second-stage solution

  – Solve sub-problem (**P4**) using the L-shaped (Algorithm 1) and obtain a solution x^*

  – Compute $(r_n^{*+}, r_n^{*-}) = \operatorname{argmin}_{r^+, r^- \in \mathbb{N}^I} \overline{Q}_M(x^*; \xi_n),\ n = 1, \ldots, N$

  # 3.2. Add more linear cuts if there are unsatisfied chance constraints

  for $n = 1, \ldots, N;\ k = 0, \ldots, K$ **do**

    $z_n^* = x^* + r_n^{*+} - r_n^{*-}$

    if $\hat{g}_{k,M}(z_n^*; \xi_n) < 1 - \delta_k$ **then**

      Add sub-gradient cut (2) and corresponding MIR inequalities (11) to the set $\{A^n z \leq b^n\}$

until $\hat{g}_{k,M}(x^* + r_n^{*+} - r_n^{*-}; \xi_n) \geq 1 - \delta_k$ *for all* n *and* k *# Stop when all constraints are satisfied*;

linear cuts added after Step 1 and 2 of Algorithm 2 might be not sufficient to approximate the chance constraints, in Step 3 we need to solve the approximate problem (**P4**) to get first- and second-stage solutions and add more cuts if these solutions do not satisfy the chance constraints. In Step 3.1, we can either solve (**P4**) by a MIP solver (e.g., CPLEX) if it is not too large, or use the L-shaped method in a large-scale setting.

Proposition 3 *Assuming that the arrival rates are always bounded from above and the support set X of first-stage solutions is finite, Algorithm 2 stops after a finite number of iterations.*

Proof. The L-shaped algorithm generates a sequence of first-stage candidates $\{x^0, x^1, \ldots\}$, and based on the properties of the optimality cuts, the algorithm stops when it finds a candidate solution that was already seen previously (Benders, 1962). Since X is finite, this implies that the algorithm must stop after a finite number of steps. In Steps 1 and 3 of Algorithm 2, for each scenario, each time when a staffing solution is infeasible, this solution is removed by sub-gradient

cuts. Since the arrival rates are bounded from above, the number of infeasible solutions (r^+, r^-) for the second-stage problem is finite, so the number of added cuts for each scenario must be finite. Therefore, Algorithm 2 converges in a finite number of iterations. ■

5 Numerical Illustrations

5.1 Algorithms and Experimental Setting

We evaluate the performance of the proposed simulation-based decomposition algorithm using three call center models of different sizes: a small one, a medium one, and a large one. We compare our approach with the algorithm presented in Chan et al. (2016), which solves (**P4**) directly via a MIP solver such as CPLEX. Problem (**P4**) can be formulated as the following deterministic equivalent problem

$$(\textbf{MIP}) \qquad \begin{cases} \min\limits_{x, r_n^+, r_n^-} & c^{\mathrm{T}} x + \dfrac{1}{N} \displaystyle\sum_{n=1}^{N} (c^+)^{\mathrm{T}} r_n^+ + (c^-)^{\mathrm{T}} r_n^- \\ \text{subject to} & A^n (x + r_n^+ - r_n^-) \le b^n, \ \forall n = 1, \ldots, N \\ & \mathcal{H}^n (x + r_n^+ - r_n^-) + \mathcal{K}^n w \le h^n, \ \forall n = 1, \ldots, N \\ & x \in X, \ r_n^+, r_n^- \in \mathbb{N}^I, \ w \ge 0. \end{cases}$$

When reporting our results, we denote Algorithm 2 by LS and the approach in which (**P4**) is solved directly by CPLEX by DE (deterministic equivalent). In our experiments with the three examples, we use the multi-cut LS (Algorithm 1). We will show in Section 5.6 that this multi-cut version outperforms the single-cut one, especially for medium and large call centers.

When running the algorithms, we use independent random numbers across the scenarios for both the first-stage and second-state simulations. When estimating subgradients, on the other hand, we use common random numbers across the two terms of the finite difference, as in Ceşik and L'Ecuyer (2008). We select step sizes $d = 1$ and $s = 1$, and take $M = 1000$ for all examples. To assess the quality of the solutions returned by the algorithms, we perform an out-of-sample evaluation of each returned solution on an independent set of scenarios. For this, we take 1000 scenarios for the small example, and 100 scenarios for the medium and large examples. We compute and report the "out-of sample" costs given by the first-stage solutions returned by the LS and DE approaches for these new sets of scenarios.

In our experiments, the cost c_i of an agent of group i is taken as an affine function of its number of skills:

$$c_i = 1 + 0.05(|\mathcal{S}_i| - 1)$$

where $|\mathcal{S}_i|$ is the cardinality of $\mathcal{S}_i$, for all i, and $c = (c_1, \ldots, c_I)^{\mathrm{T}}$. For the costs of adding or removing agents, we consider three cases, labeled R1, R2, and R3, as defined in Table 1.

Test case	c^+	c^-
R1	$2c$	$0.5c$
R2	$1.5c$	$0.75c$
R3	$1.1c$	$0.9c$

Table 1: Costs of adding and removing agents

The arrival rate λ_k for call type k (for the entire period) is the realization of a random variable Λ_k of the form $\Lambda_k = \xi_k \beta_k$ (a product of two random variables), where the realization of ξ_k is unveiled at the beginning of the second stage, while β_k remains unknown. For our illustrations, we take simple choices of distributions: we suppose that ξ_k has a truncated normal distribution with parameters that generally depend on k and that β_k follows a symmetric triangular distribution of mean and mode 1, minimum 0.9, and maximum 1.1 (see Avramidis et al., 2004). The normal distribution is truncated to satisfy the assumptions of Proposition 3.

The experiments were conducted on a machine running Debian 8 with Intel(R) Xeon(R) E5620 CPUs running at 2.40GHz. The computer has 8 physical CPUs and 98GB of memory. The simulations were made using the *ContactCenters* simulation software (Buist and L'Ecuyer, 2005), developed in Java with the SSJ simulation library (L'Ecuyer et al., 2002). The algorithms were coded in MATLAB and linked to IBM ILOG CPLEX 12.6 optimization routines under default settings. To speed up the computations, the steps of performing simulations and adding sub-gradient cuts for each scenario were run in parallel using the 8 physical CPUs.

5.2 Example 1: A Small Call Center

We first consider a small call center with $K = 2$ call types and $I = 2$ agent groups, with $\mathcal{S}_1 = \{1\}$ and $\mathcal{S}_2 = \{1, 2\}$. This small example will permit us to use a larger number of scenarios than the larger ones. We assume that for the two call types: (i) each caller abandons with probability 2% if it has to wait, (ii) patience times are exponential with means 10 and 6 minutes, (iii) the "mean" arrival rate ξ_k follows a normal distribution with means 100 and 70 calls per hour, 15% standard deviations from the means, and truncated to intervals [75, 125] and [52.5, 87.5], respectively, and (iv) all service times are exponential with means 10 and 7.5 minutes. The length of the period is one hour. The parameters in the SL constraints are $\tau_1 = \tau_2 = \tau_0 = 120$ (seconds), $l_1 = l_2 = 80\%$, and $l_0 = 85\%$. For each case (R1, R2, and R3), we generate 5 independent sets of 100, 200, 300, 400, 500 scenarios. The parameters α_k for the initial constraints with the fluid model are taken as $\alpha_1 = \alpha_2 = 1$.

For DE, the MIP problem **(MIP)** is typically very large because of the large number of scenarios, and CPLEX cannot find an optimal solution even with a time budget of several hours. So we set the time limit to 200 seconds and the optimal gap to 0.05%. The first step (Initialization) in Algorithm 2 takes about 85 and 400 seconds for the instances with 100 and 500 scenarios, respectively. This step is the same for LS and DE, so in the tables we only report the computing times for the remainder, i.e., for solving the two-stage stochastic linear programs and to generate

N	Methods	R1			R2			R3		
		Cost	Time (s)	Out of sample cost	Cost	Time (s)	Out of sample cost	Cost	Time (s)	Out of sample cost
100	LS	32.72	**73**	33.13	32.33	**148**	32.25	31.39	**76**	**31.60**
	DE	**32.65**	275	**33.03**	**32.31**	828	**32.15**	31.39	277	31.61
200	LS	32.74	**296**	33.13	32.08	**298**	32.21	31.55	**315**	31.61
	DE	32.74	693	33.13	**32.06**	697	**32.12**	**31.52**	350	**31.54**
300	LS	33.00	**456**	33.13	32.18	**449**	32.22	31.56	**476**	31.61
	DE	**32.96**	838	**33.03**	**32.15**	839	**32.13**	**31.55**	1290	31.61
400	LS	32.58	**611**	33.15	32.09	**896**	32.22	31.43	**633**	31.61
	DE	32.58	982	33.15	**32.06**	981	**32.12**	**31.42**	1507	31.61
500	LS	32.82	**771**	33.13	32.02	**765**	32.21	31.29	**830**	31.61
	DE	**32.81**	1133	**33.03**	**32.01**	1131	**32.12**	31.29	1170	31.61

Table 2: Value of the best solution found for (**P4**) (Cost), total CPU time (in seconds, excluding the initialization), and cost of the retained first-stage solution of (**P4**) estimated out-of-sample, for the small call center

more cuts by simulation. The LS method needs only a few seconds to return a solution. The rest of the time is for the simulation.

The results obtained by LS and DE for the three cost structures R1, R2, and R3 are reported in Table 2. The smallest costs and CPU times are in bold. While the objective values returned by both approaches are similar, DE gives slightly better costs in 11 instances out of 15. On the other hand, LS runs much faster than DE for all instances. The L-shaped method returns a solution in a few seconds, while CPLEX always exceeds the time budget of 200 seconds. In the out-of-sample evaluation, the two methods return solutions having the same cost in 5/15 instances, DE returns a less expensive solution in 9/15 instances, and it returns a slightly worse cost in one instance. In all cases, the out-of-sample costs given by the two methods are quite close in value. Overall, we find that LS is highly preferable for this example, because it returns its solution much faster, and the solution is almost never worse than with DE.

5.3 Example 2: A Medium-Size Call Center

We now consider a medium-size call center with $K = 6$ call types and $I = 8$ agent groups. We assume that (i) the callers do not abandon immediately in case they have to wait, (ii) patience times are exponential with means between 36 and 52 minutes, (iii) for the different k, we suppose that ξ_k follows a normal distribution with mean from 0.45 to 9.15 calls per minute and standard deviation which is 10% of the mean, truncated to 2.5 standard deviations on each side of the mean, and (iv) all service times have a lognormal distribution with mean between 5.1 and 11.3 minutes. The length of the period is 10 hours. We take $\tau_k = 120$ (seconds) and $l_k = 80\%$ for $k = 0, \ldots, K$. We try two sets of targets for the chance constraints: (i) $1 - \delta_0 = 85\%$ and $1 - \delta_k = 80\%$, $k = 1, \ldots, K$, and (ii) $1 - \delta_0 = 95\%$ and $1 - \delta_k = 90\%$ for $k = 1, \ldots, K$. The parameters α_k in the fluid model are taken as $1, 4, 1, 1.2, 1, 3$. These values were adjusted manually to ensure that the initial constraints were removing the non-concave regions of the probability functions.

			Cost		CPU time (hour)		Out-of-sample cost	
$(1-\delta_k, 1-\delta_0)$	Cases	N	LS	DE	LS	DE	LS	DE
(0.80, 0.85)	R1	20	186.90	186.90	**0.26**	0.54	188.25	**188.11**
		50	**188.10**	188.15	**1.95**	3.43	**188.02**	188.09
		70	**184.35**	184.63	**1.55**	3.94	**188.6**	188.91
	R2	20	**179.39**	179.47	**0.74**	3.61	**186.99**	187.26
		50	179.94	**179.90**	**1.87**	3.10	185.60	**185.22**
		70	**183.86**	183.87	**3.32**	5.45	**183.17**	183.32
	R3	20	**180.10**	180.13	**1.44**	2.57	**187.41**	188.10
		50	**177.43**	177.48	**1.61**	5.25	184.15	184.15
		70	**175.31**	175.41	**3.57**	10.08	**183.32**	183.64
(0.90, 0.95)	R1	20	191.43	**191.34**	**1.31**	2.94	194.68	**194.65**
		50	**193.64**	193.78	**1.17**	2.69	**194.32**	194.57
		70	**191.16**	191.17	2.39	**2.33**	**195.32**	195.46
	R2	20	185.77	**185.75**	**0.37**	0.87	**193.20**	193.34
		50	**186.40**	186.46	**2.07**	3.84	**191.21**	191.32
		70	**190.33**	190.34	**1.44**	4.63	**189.71**	189.89
	R3	20	185.03	**185.00**	**2.06**	4.00	195.62	**194.89**
		50	**183.21**	183.30	**2.54**	4.43	190.72	190.72
		70	**180.47**	180.61	**3.38**	10.65	189.28	189.28

Table 3: Value of best solution found for (**P4**) (Cost), CPU time (in hours, excluding initialization), and cost of retained first-stage solution estimated out-of-sample, for the medium-size call center

Since the simulation here is more expensive than for the small call center of the previous example, we only consider instances with less than 100 scenarios. For each cost structure, we independently generate instances of 20, 50, 70 scenarios and we use the sample size $M = 1000$ to estimate the chance constraints. We solve each instance and report the corresponding first-stage solutions. For the out-of-sample validation, we use 100 independent scenarios. We set a time budget of 10 minutes for CPLEX.

Table 3 reports the results. As in the previous example, the smallest costs and shortest CPU times are indicated in bold. Note that Step 1 in Algorithm 2) takes about 0.14, 0.22, and 0.38 hours for the instances with 20, 50, and 70 scenarios, respectively. The solutions returned by the two methods have similar costs, both in-sample and out-of-sample, although LS is slightly better more often than DE. It is also significantly faster. We also see that better solutions are obtained when increasing the number of scenarios from 20 to 70 for the cost structures R2 and R3, but not for R1, for which the difference between c^+ and c^- is larger.

In Table 4, we report the first-stage solutions, the first-stage costs as well as the averages of the numbers of added or removed agents for the three cases with $N = 70$ scenarios. As for the small call center, we see that the first-stage costs under R1 and R2 are higher than under R3, and the average value of r^+ under R1 and R2 is smaller than under R3.

$(1-\delta_k, 1-\delta_0)$	Case	Algorithm	x^{T}	c^{T} x	Average r^+	Average r^-
(0.80,0.85)	R1	LS	(33, 26, 88, 6, 0, 0, 4, 11)	181.30	4.11	10.64
		DE	(34, 26, 88, 6, 0, 0, 5, 10)	182.40	3.78	11.27
	R2	LS	(34, 26, 92, 6, 0, 0, 6, 10)	187.70	4.30	14.40
		DE	(33, 26, 92, 5, 0, 0, 6, 11)	186.60	3.99	11.34
	R3	LS	(33, 23, 84, 7, 0, 0, 3, 4)	165.90	15.36	9.24
		DE	(33, 23, 84, 8, 0, 0, 2, 5)	167.10	14.06	8.67
(0.90,0.95)	R1	LS	(37, 25, 91, 4, 3, 0, 6, 7)	186.70	4.71	10.69
		DE	(36, 26, 92, 4, 2, 0, 6, 7)	186.55	4.76	10.63
	R2	LS	(32, 27, 93, 7, 2, 0, 4, 12)	190.90	5.21	11.04
		DE	(32, 27, 94, 6, 2, 0, 4, 13)	192.00	4.63	11.20
	R3	LS	(32, 24, 86, 8, 0, 0, 3, 5)	170.15	17.33	10.66
		DE	(32, 24, 86, 8, 0, 0, 3, 5)	170.15	17.03	10.11

Table 4: First-stage solutions, first-stage costs and average number of additions and removals of agents for $N = 70$ for the medium call center

5.4 Example 3: A Larger Call Center

We now consider a larger call center with $K = 20$ call types and $I = 15$ agent groups. We assume that (i) the callers abandon with probability 0.1 in case they have to wait, (ii) all patience times are exponential with means 6 minutes (iii) for each call type k, the arrival rate ξ_k follows a normal distribution with mean from 130 to 260 calls per hour and standard deviations which is 10% of the mean, truncated to 2.5 standard deviations, and (iv) all service times are exponential with means 7.5 minutes. We take $\tau_k = \tau_0 = 20$ (seconds), $l_k = 50\%$ for $k = 1, \ldots, K$, and $l_0 = 80\%$. The length of the period is one hour. For the chance constraints, we try (i) $1 - \delta_0 = 85\%$ and $1 - \delta_k = 80\%$, $k = 1, \ldots, K$, and (ii) $1 - \delta_0 = 95\%$ and $1 - \delta_k = 90\%$ for $k = 1, \ldots, K$. For each cost structure R1, R2 and R3, we test LS and DE with 20, 50, and 70 scenarios. For DE, we give CPLEX a time budget of 10 minutes and a MIP gap of 0.05%. We also take $\alpha_k = 1$ for all k.

Table 5 reports the results. Step 1 of Algorithm 2 takes from 1.6 to 4.0 hours for 20 to 70 scenarios, respectively. Both in-sample and out-of-sample, the costs of the retained solutions are slight better (but not much) for LS than for DE. LS also requires much less CPU time for all instances. The out-of-sample costs are also improved when we increase the number of scenarios.

Note that the computing time for one iteration of Step #3 of Algorithm 2 for the LS and the DE can be approximated as $(\nu_{\mathrm{LS}} + \nu)$ and $(\nu_{\mathrm{DE}} + \nu)$, respectively, where ν_{LS} stands for the total computing time to solve (**P4**) by the LS method (Algorithm 2), ν_{DE} stands for the total computing time required by CPLEX to solve **(MIP)**, ν is the CPU time required to perform simulation and add more sub-gradient cuts for unsatisfied chance constraints (Step 2.2 in Algorithm 2). For the medium and large examples, ν_{LS} is very small (a few seconds, see Table 8 below) compared to ν_{DE} (set as 10 minutes). The number of iterations at Step #3 is always smaller for LS than for DE, and this is why LS is faster than DE in all instances.

Table 6 reports the first-stage solutions, first-stage costs and the average numbers of added or removed agents for R1, R2, R3. As for the small and medium call centers, we observe that in

			Cost		CPU time (hour)		Out-of-sample cost	
$(1-\delta_k, 1-\delta_0)$	Case	N	LS	DE	LS	DE	LS	DE
(0.80, 0.85)	R1	20	**156.62**	156.75	**1.08**	5.20	159.77	**158.64**
		50	**157.13**	157.20	**2.44**	7.98	**158.59**	159.39
		70	**157.66**	158.65	**2.29**	9.90	**158.52**	158.75
	R2	20	**157.47**	158.31	**1.07**	5.24	**157.85**	158.50
		50	**156.54**	156.67	**1.98**	7.27	**156.98**	157.07
		70	**154.86**	156.11	**2.67**	6.34	**156.78**	156.96
	R3	20	**161.61**	164.37	**1.12**	4.89	**158.43**	157.58
		50	**154.65**	154.86	**2.37**	10.48	**158.03**	158.09
		70	**155.72**	159.18	**2.76**	9.01	**158.02**	159.43
(0.90, 0.95)	R1	20	170.23	**170.18**	**0.04**	1.17	174.85	**174.83**
		50	**171.55**	171.68	**0.41**	1.93	**174.44**	174.66
		70	**172.44**	172.61	**0.82**	1.82	**174.57**	174.78
	R2	20	172.02	**171.95**	**0.29**	1.28	**173.82**	173.92
		50	**169.97**	170.01	**0.16**	0.96	**172.46**	172.57
		70	**169.63**	169.77	**0.58**	1.19	**172.68**	172.70
	R3	20	**168.37**	168.44	**0.40**	3.95	175.56	**175.07**
		50	**167.17**	167.64	**0.96**	5.19	**173.60**	174.79
		70	**167.38**	167.80	**1.19**	5.04	**171.02**	171.77

Table 5: Value of the best solution found for (**P4**) (Cost), CPU time (in hours, excluding the initialization), and cost of retained solution estimated out-of-sample, for the large call center

all three cases, LS generally gives a slightly lower first-stage cost than DE. The costs for R1 and R2 are larger than for R3, and we also obtain a larger first-stage cost when the SL targets are higher. Moreover, the average numbers of added or removed agents is smaller for R1 and R2 than for R3.

5.5 Value of a Stochastic Solution

Some may argue that the two-stage stochastic model considered in this paper is too much work, in particular with large-scale call centers, as the model involves a set of solutions instead of one solution as in one-stage models. To show that this more complicated two-stage stochastic model is worthwhile, we evaluate its relevance using the notion of *value of a stochastic solution* (VSS). For this, we solve a much simpler problem in which all the random variables are replaced by their expected values. In our context, it means that we solve the following one-stage staffing optimization problem, called the *mean value problem*, in which the random factor ξ is replaced by its expectation $\bar{\xi} = \mathbb{E}[\xi]$:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^{\mathrm{T}} x \\ \text{subject to} \quad & \mathbb{P}[\mathcal{S}_k(x) \geq l_k \mid \bar{\xi}] \geq 1 - \delta_k, \quad k = 0, \ldots, K, \\ & x \geq 0 \text{ and integer.} \end{aligned} \tag{16}$$

In general, there is no reason to believe that a solution $x(\bar{\xi})$ to this problem is close to a solution to the recourse problem (**P1**), and the VSS is a concept to measure how bad a decision $x(\bar{\xi})$ is compared to a solution of the more realistic recourse model (**P1**).

$(1-\delta_k, 1-\delta_0)$	Models	Algorithms	x^{T}	c^{T} x	Averaged r^+	Averaged r^-
(0.80,0.85)	R1	LS	(20, 0, 3, 15, 0, 0, 21, 12, 15, 7, 12, 5, 5, 6, 8)	156.05	1.44	2.92
		DE	(21, 0, 2, 16, 0, 4, 22, 13, 14, 10, 11, 2, 4, 5, 6)	157.50	1.35	3.50
	R2	LS	(20, 0, 5, 15, 0, 0, 22, 12, 13, 10, 11, 6, 5, 5, 5)	156.10	1.33	3.87
		DE	(19, 0, 5, 16, 0, 0, 23, 12, 11, 7, 12, 4, 8, 6, 7)	157.05	1.70	4.39
	R3	LS	(19, 0, 6, 17, 0, 0, 21, 13, 15, 7, 11, 5, 3, 2, 7)	151.85	4.94	3.50
		DE	(19, 0, 4, 16, 0, 0, 22, 16, 13, 7, 12, 0, 4, 6, 7)	151.95	4.58	2.94
(0.90,0.95)	R1	LS	(21, 0, 7, 18, 0, 3, 23, 11, 15, 6, 11, 6, 7, 3, 10)	170.35	1.93	4.17
		DE	(20, 0, 8, 18, 0, 3, 23, 11, 16, 6, 12, 6, 6, 3, 10)	170.40	1.64	4.59
	R2	LS	(22, 0, 6, 17, 0, 1, 22, 14, 15, 7, 12, 3, 7, 4, 10)	168.00	2.79	4.83
		DE	(22, 0, 6, 17, 0, 1, 22, 13, 15, 7, 13, 3, 7, 4, 10)	168.00	2.76	4.60
	R3	LS	(19, 0, 6, 18, 0, 0, 22, 14, 15, 4, 13, 4, 4, 4, 10)	159.60	10.99	7.14
		DE	(20, 0, 6, 18, 0, 0, 22, 14, 15, 4, 13, 4, 5, 4, 10)	162.00	10.16	7.87

Table 6: First-stage solutions, first-stage costs and average number of added or removed agents for $N = 70$, for the large call center

To compute the VSS, we first solve the *mean value problem* (16) using the SAA method with sample size $M = 1000$, to obtain a solution $x(\bar{\xi})$. The VSS can then be computed as the gap between the (out-of-sample) cost of a solution obtained by solving (**P2**) and the cost of the solution $x(\bar{\xi})$ when used in the two-stage model. We compute the VSS for the three instances of 70 scenarios with two sets of targets (0.80,0.85) and (0.90,0.95), as in the previous sections. For the costs of the recourse problems, we use those obtained by the LS approach, noting that the costs given by the DE are also quite similar. Table 7 reports the VSS as well as the percentage of increase (denoted by "% of increase" in the table) of the cost when we use the *mean value problem*, compared with the cost of our best solution to the two-stage problem. The reported VSS and their relative values (the percentages) are quite significant, especially with R3 and the larger targets (0.90, 0.95). We also observe a VSS increase from R1 to R3, and from moderately low targets (0.80, 0.85) to higher ones (0.90, 0.95). This shows that the cost of ignoring uncertainty in choosing a staffing decision is significant.

5.6 A Comparison of the Single-cut and Multi-cut LS Approaches

We provide a brief comparison of the performance of the multi-cut and single-cut L-shaped approaches on our three call center examples. For the multi-cut approach we choose $L = N$ (we generate cuts for each scenario), as in our context the number of scenarios is not large and choosing $L = N$ reduces the number of iterations in Algorithm 1. We try the three cost structures R1, R2, and R3 for the recourse. We also set a limit of 300 iterations per call of

		Medium example		Large example	
$(1-\delta_k, 1-\delta_0)$	Cases	VSS	% of increase	VSS	% of increase
(0.80,0.85)	R1	4.69	2.55%	8.79	5.58%
	R2	12.86	7.00%	10.28	6.64%
	R3	17.07	9.74%	13.01	8.35%
(0.90,0.95)	R1	6.89	3.60%	10.25	5.94%
	R2	15.37	8.08%	11.98	7.06%
	R3	19.88	11.02%	15.16	9.06%

Table 7: VSS for the medium and large examples

Algorithm 1. Table 8 reports the average number of iterations and the average CPU times in Algorithm 1 per call to this algorithm, with each of the two LS approaches. Again, the smallest numbers are in bold. The symbol "–" indicates that the corresponding approach failed to converge within 300 iterations. We find that for all call center sizes, the multi-cut approach requires fewer iterations. The CPU time is slightly larger with the multi-cut for the small call center (and also increases with the number N of scenarios, because there are then more constraints in the master problem), but becomes much smaller than for the single-cut when the size of the model increases. For the largest model, the single-cut approach fails to converge within 300 iterations in all instances, while the multi-cut converges in about 20 to 60 iterations, and the average CPU times are reasonable (20 to 200 seconds). The results clearly show the superiority of the multi-cut approach for our simulation-based decomposition algorithm, in particular for large instances.

Case	# scenarios		Small call center			Medium call center			Large call center		
			300	600	800	20	50	70	20	50	70
R1	single-cut	# iterations	7.4	6.6	6.6	71.2	111.5	99.6	–	–	–
		CPU time (s)	**3.6**	**6.5**	**8.9**	59.5	98.9	190.0	–	–	–
	multi-cut	# iterations	4.7	4.4	3.9	26.2	26.2	24.0	54.2	26.6	23.6
		CPU time (s)	4.3	17.9	20.2	**17.7**	**21.1**	**29.5**	**156.6**	**31.6**	**29.7**
R2	single-cut	# iterations	9.1	8.5	9.3	79.2	82.5	98.4	-	-	-
		CPU time (s)	**7.5**	**15.7**	**20.4**	65.2	95.6	181.3	-	-	-
	multi-cut	# iterations	8.1	7.9	7.5	25.3	24.5	23.7	62.2	55.6	58.1
		CPU time (s)	11.7	42.3	82.4	**16.3**	**21.0**	**27.9**	**70.1**	**65.3**	**72.3**
R3	single-cut	# iterations	8.2	9.4	9.1	58.8	72.3	70.2	-	-	-
		CPU time (s)	**9.6**	**16.3**	**26.6**	51.2	79.3	145.2	-	-	-
	multi-cut	# iterations	7.5	8.1	8.9	19.3	20.1	18.9	43.4	38.2	35.7
		CPU time (s)	11.9	48.2	98.7	**13.3**	**17.2**	**21.2**	**53.3**	**45.3**	**56.2**

Table 8: Comparison of the single-cut and multi-cut approaches

6 Conclusion

We have proposed and tested a simulation-based SAA method combined with a decomposition algorithm for staffing optimization under arrival rate uncertainty. The problem is formulated as a two-stage stochastic program with integer recourse. We reported numerical results based on call center models of three different sizes. Our results show that the decomposition approach outperforms a direct approach that does not use decomposition to solve the approximating

MIP problem, especially for the large call center example. It provides good-quality solutions in reasonable time.

These results open several interesting directions for future research, e.g., the extension of the method from staffing to scheduling problems. The proposed methodology might also be useful for other similar workforce management problems, such as staffing and scheduling in hospitals, clinics, and retail stores, for example, and especially for applications in which the constraints are constructed based on complex queuing models, for which the performance needs to be approximated by simulation.

Acknowledgment

This work has been supported by a Canada Research Chair, an Inria International Chair, and a Hydro-Québec research grant to P. L'Ecuyer, by NSERC Discovery Grants to F. Bastin and P. L'Ecuyer, and by scholarships from the CIRRELT, DIRO and Université de Montréal to T.A. Ta. We benefited from valuable discussions with Tien Mai from Singapore-MIT Alliance for Research and Technology (SMART).

References

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

J. Atlason, M. A. Epelman, and S. G. Henderson. Optimizing call center staffing using simulation and analytic center cutting plane methods. *Management Science*, 54(2):295–309, 2008.

A. N. Avramidis and P. L'Ecuyer. Modeling and simulation of call centers. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 144–152. IEEE Press, 2005.

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

A. N. Avramidis, W. Chan, and P. L'Ecuyer. Staffing multi-skill call centers via search methods and a performance approximation. *IIE Transactions*, 41(6):483–497, 2009.

A. N. Avramidis, W. Chan, M. Gendreau, P. L'Ecuyer, and O. Pisacane. Optimizing daily agent scheduling in a multiskill call centers. *European Journal of Operational Research*, 200 (3):822–832, 2010.

A. Bassamboo, J. M. Harrison, and A. Zeevi. Design and control of a large call center: Asymptotic analysis of an LP-based method. *Operations Research*, 54(3):419–435, 2006. ISSN 0030-364X.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, NY, USA, second edition, 2011.

E. Buist and P. L'Ecuyer. A Java library for simulating contact centers. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 556–565. IEEE Press, 2005.

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

W. Chan. *Optimisation des horaires des agents et du routage des appels dans les centres d'appels*. PhD thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Canada, 2013.

W. Chan, G. Koole, and P. L'Ecuyer. Dynamic call center routing policies using call waiting and agent idle times. *Manufacturing & Service Operations Management*, 16(4):544–560, 2014.

Wyean Chan, Thuy Anh Ta, Pierre L'Ecuyer, and Fabian Bastin. Two-stage chance-constrained staffing with agent recourse for multi-skill call centers. In *Proceedings of the 2016 Winter Simulation Conference*, pages 3189–3200, Piscataway, NJ, USA, 2016. IEEE Press.

N. Channouf, P. L'Ecuyer, A. Ingolfsson, and A. N. Avramidis. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science*, 10(1):25–45, 2007.

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

N. Gans, H. Shen, Y.-P. Zhou, N. Korolev, A. McCord, and H. Ristock. Parametric forecasting and stochastic programming models for call-center workforce scheduling. *Manufacturing and Service Operations Management*, 17(4):571–588, 2015.

L. V. Green, P. J. Kolesar, and J. Soares. An improved heuristic for staffing telephone call centers with limited operating hours. *Production and Operations Management*, 12:46–61, 2003.

I. Gurvich, J. Luedtke, and T. Tezcan. Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach. *Management Science*, 56(7):1093–1115, 2010.

J. M. Harrison and A. Zeevi. A method for staffing large call centers based on stochastic fluid models. *Manufacturing and Service Operations Management*, 7(1):20–36, 2005.

S. Helber and K. Henken. Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials. *OR Spectrum*, 32:109–134, 2010. ISSN 0171-6468.

R. Ibrahim, P. L'Ecuyer, N. Régnard, and H. Shen. On the modeling and forecasting of call center arrivals. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 256–267. IEEE Press, 2012.

R. Ibrahim, H. Ye, P. L'Ecuyer, and H. Shen. Modeling and forecasting call center arrivals: A literature study and a case study. *International Journal of Forecasting*, 32(3):865–874, 2016.

O. Jouini, G. Koole, and A. Roubos. Performance indicators for call centers with impatient customers. *IIE Transactions*, 45(3):341–354, 2013.

G. Koole. *Call Center Optimization.* MG books, Amsterdam, 2013.

P. L'Ecuyer and E. Buist. Variance reduction in the simulation of call centers. In *Proceedings of the 2006 Winter Simulation Conference*, pages 604–613. IEEE Press, 2006.

P. L'Ecuyer, L. Meliani, and J. Vaucher. SSJ: A framework for stochastic simulation in Java. In E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 234–242. IEEE Press, 2002.

S. Liao, C. van Delft, G. Koole, and O. Jouini. Staffing a call center with uncertain non-stationary arrival rate and flexibility. *OR Spectrum*, 34:691–721, 2012.

S. Liao, C. van Delft, and J. P. Vial. Distributionally robust workforce scheduling in call centers with uncertain arrival rates. *Optimization Methods and Software*, 28(3):501–522, 2013.

G. L. Nemhauser and L. A. Wolsey. A recursive procedure to generate all cuts for 0–1 mixed integer programs. *Mathematical Programming*, 46(1-3):379–390, 1990.

B. Oreshkin, N. Régnard, and P. L'Ecuyer. Rate-based daily arrival process models with application to call centers. *Operations Research*, 64(2):510–527, 2016. doi: 10.1287/opre.2016.1484.

A. Pot, S. Bhulai, and G. Koole. A simple staffing method for multi-skill call centers. *Manufacturing and Service Operations Management*, 10:421–428, 2008.

T. R. Robbins and T. P. Harrison. A stochastic programming model for scheduling call centers with global service level agreements. *European Journal of Operational Research*, 207(3):1608–1619, 2010.

T. A. Ta, T. Mai, F. Bastin, and P. L'Ecuyer. On a two-stage discrete stochastic optimization problem with stochastic constraints and nested sampling. *Submitted for publication*, 2018.

R. B. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management*, 7(4):276–294, 2005.