8-2022

Joint chance-constrained staffing optimization in multi-skill call centers

Tien Thanh DAM

Thuy Anh TA

Tien MAI
Singapore Management University, atmai@smu.edu.sg

Journal of Combinatorial Optimization
https://doi.org/10.1007/s10878-021-00830-1

Joint chance-constrained staffing optimization in multi-skill call centers

Tien Thanh Dam[1] · **Thuy Anh Ta**[1] · **Tien Mai**[2]

Abstract
This paper concerns the staffing optimization 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 (QoS) to customers. We consider a staffing problem in which joint chance constraints are imposed on the QoS of the day. Our joint chance-constrained formulation is more rational capturing the correlation between different call types, as compared to separate chance-constrained versions considered in previous studies. We show that, in general, the probability functions in the joint-chance constraints display S-shaped curves, and the optimal solutions should belong to the concave regions of the curves. Thus, we propose an approach combining a heuristic phase to identify solutions lying in the concave part and a simulation-based cut generation phase to create outer-approximations of the probability functions. This allows us to find good staffing solutions satisfying the joint-chance constraints by simulation and linear programming. We test our formulation and algorithm using call center examples of up to 65 call types and 89 agent groups, which shows the benefits of our joint-chance constrained formulation and the advantage of our algorithm over standard ones.

Keywords Call center · Staffing optimization · Joint chance constraint · Cutting plane · Concave-identification

1 Introduction

This paper concerns the management of staffing in a *call center*, which can be generally defined as a central office used for receiving or transmitting customers' requests by

✉ Thuy Anh Ta
anh.tathuy@phenikaa-uni.edu.vn

1 ORLab, Faculty of Computer Science, Phenikaa University, Yen Nghia, Ha Dong, Hanoi, Vietnam

2 School of Computing and Information Systems, Singapore Management University, Singapore, Singapore

telephone. In a call center, an inbound call is used for the purposes of purchase, troubleshooting, repair request, emergency call, etc., while an outbound call is for the purposes of consulting, recruiting, telemarketing, market research, and so on. Call centers are quite popular in society and can be found in customer care services, fire alarms, telemarketing, etc. The call center industry is growing and playing an important role in society. For instance, in the United States, the number of call center agents rose from 2.1 million in 2004 to 2.7 millions in 2016 with an estimated annual salary cost of US $95.2 billion (according to the Bureau of Labor Statistics in 2016 (https://www.bls.gov/oes/2016/may/naics5_561420.htm). Nowadays, with the development of technology, the call center provides services not only on phones but also via email and chat. For more details about call centers, we refer the reader to Gans et al. (2003), Wallace and Whitt (2005), Ingolfsson et al. (2010).

The aim of this paper is to develop mathematical models and optimization algorithms for workforce planning in telephone call centers, for the purpose of improving their operations. More precisely, we study the staffing optimization problem in multiskill call centers, in which we aim at minimizing an operating cost while delivering a high quality of service (QoS) to customers. This is one of the essential problems in the management of a call center. We propose a new optimization model in which joint chance constraints are imposed on some performance measures, and develop a new optimization algorithm to solve the problem practically. We also provide numerical experiments to show the advantage of our model and algorithm.

In call center systems, *performance measures* are used to access the *quality of service* (QoS) and efficiency of a call center. The main purpose of these performance measures is to ensure that the call center is meeting its goal and objectives. *Service level* (SL) is one of the most popular QoS measures. It can be defined as the fraction of calls that are answered within an acceptable waiting time. The constraint on the SL is mostly stated as s percent of calls answered within τ seconds, where τ is a given parameter. The SL can be measured and controlled separately by call types, or in an aggregated day. In practice, SL can be defined as an expectation over an infinite time horizon, or as a random variable over a given time period. In the latter case, one may use chance constraints to ensure that the probability of meeting an SL requirement is above a certain level. Various closed-form formulas have been proposed to compute SL values in different settings (Buist and L'Ecuyer 2005; Jouini et al. 2012). However, in the case of multi-skill call centers, it is not possible to exactly calculate the SLs. Instead, one needs to use simulation to approximate them (Çezik and L'Ecuyer 2008).

There are many problems that need to be considered in the management of a call center. One of them is the staffing optimization problem that deals with a decision on selecting a number of agents of each skill group at each period of the day to satisfy QoS constraints while minimizing an operational cost. One might be also interested in a scheduling problem (Avramidis et al. 2010), in which a set of admissible shift schedules is first specified and one needs to choose a number of agents of each group in each shift. The number of decision variables and data in a scheduling problem is a combination of work shifts, breaks, training time, etc, so it is much larger as compared to the staffing optimization one. Call routing is also an optimization issue (Chan et al. 2014; Kooley et al. 2015). In this paper, we focus on the staffing optimization problem.

To ensure the target of SL over finite duration we can use chance constraints. These constraints are adequate when the performance is measured over a short time interval and QoS measures are random variables in a given period off time. In addition, the chance-constrained approach does not require that the decisions are feasible for (almost) every outcome of random parameters, but ensures that the probability of satisfying a certain set of SL constraints is above a certain level. A general and popular way to deal with chance-constrained programs is to build sample approximations of chance constraints using the SAA method (Nemirovski and Shapiro 2006b). The advantages of the chance constraints in the context of the call center staffing optimization have been discussed thoroughly in some recent studies (Chan et al. 2014, 2016; Ta et al. 2019), to which we refer the reader for more details. Separate chance constraints widely-used in previous work (Chan et al. 2014, 2016; Ta et al. 2019). The use of such constraints would however be problematic if the SL values from different call types are strongly correlated and a joint chance-constrained formulation could be used to overcome this issue. To illustrate the advantage of joints chance constraints over separate chance constraints, we give the following example. Suppose that X and Y are two random variables following a uniform distribution in [0, 1] such that $X = 1 - Y$. Clearly, $\mathbb{P}[X \geq 0.5] = \mathbb{P}[Y \geq 0.5] = 0.5$. However, the joint probability value $\mathbb{P}[X \geq 0.5 \ \& \ Y \geq 0.5]$ is equal to zero. Thus, the joint probability value would be much lower than each individual probability value if the random variables are highly correlated. This is also the case in the context of multi-skill centers with SL constraints, as we will show later in the experiential section that a joint probability value can be very small (close to zero) even though all the individual probability values are larger than 0.8.

Optimization problems with joint chance constraint have received much attention in the operations research and management science literature. Nemirovski and Shapiro (2006a) proposes a general class of convex conservative approximations of chance constrained problems. They also create a convex and efficiently solvable large deviation-type approximation of chance-constrained problems, known as the "Berstein approximation". Zan et al. (2014) consider the staffing problem of large-scale service systems with multiple customer classes and multiple dedicated server pools. They obtain asymptotically optimal solutions to the staffing problem by using the *Janssen–Van Leeuwaarden–Zwart* bounds. Bonferroni approximation is one of the most popular approximations of joint chance constraints, which replaces the joint chance constraint with a set of single-chance constraints such that their sum does not exceed a certain threshold. Some variations of Bonferroni approximation for joint chance constraint have been recently proposed to solve different problems (Xie et al. 2017; Singh and Watson 2019). Besides, there are several studies using joint chance constraint for problems relating to call centers. Gurvich et al. (2010) considers the staffing problem with uncertain demand forecasts in multi-skill call centers. They propose a two-step solution for the staffing problem under joint chance constraints. In step 1, they solve a random static planning problem (RSPP), i.e., they aim to obtain a staffing vector and a set of arrival rate vectors that minimize the total staffing cost and meet some probability targets. In step 2, they use the staffing vector obtained at step 1 as the initial solution and perform a simulation-based search with a fixed routing rule to find an optimal staffing solution that is feasible to the joint chance constraints. Excoffier et al. (2015)

propose a one-stage stochastic program involving joint chance constraints as a solution to deal with a staffing and shift-scheduling problem. Excoffier et al. (2016) then construct a stochastic programming-based approach to deal with arrival rate uncertainty in a shift-scheduling problem. This approach is the combination of two steps: (1) reformulating the joint chance constraint program into a deterministic program with a set of non-linear terms and (2) building a numerical representation for these non-linear terms.

The staffing optimization problems in multi-skill contact centers is challenging as it involves non-linear constraints that can only be approximated by simulation. The literature has seen a number of studies making use of simulation and linear programming to practically solve the problem. Atlason et al. (2004) propose a method that combines cut generation and linear programming to solve a scheduling problem in a single call type and single-skill call center with long-term expected SL constraints. Çezik and L'Ecuyer (2008) adapt the cutting plane method to solve large-size staffing problems in a single time period for multi-skill call centers and Atlason et al. (2008) have also developed a cutting plane method in conjunction with simulation and analytic call centers to solve the scheduling problem with more factors such as work requirements and shift construction. Another algorithm based on the cutting plane method to solve the scheduling problem in multiple time periods and constraints on work schedules is developed in Avramidis et al. (2010). Some other algorithms based on the cutting plane method and simulation have been recently proposed to solve staffing problems under separate chance constraints with arrival rate uncertainty (Chan et al. 2016; Ta et al. 2020, 2019). It is important to note here that the use of the cutting plane method is supported by the observation that the QoS functions often display "*S-shaped*" forms and good solutions typically belong to the concave regions of the QoS functions. Thus, to make sure the cutting plane returns good staffing/scheduling solutions, one needs to well eliminate the non-concave parts. This is a crucial issue when using the cutting plane method in the context.

Contributions: In this work, we consider a joint chance-constrained staffing optimization model in multi-skill centers. As being said, this model can take advantage of the chance-constrained approach over models based on classical long-term expected SLs. On the other hand, our joint chance-constrained model is also advantageous in accounting for the correlation between different call types. Nevertheless, it is well-known in the literature that joint chance-constrained problems are typically more difficult to solve, even numerically (Ahmed and Shapiro 2008). To address the challenge, we examine some essential properties of the joint chance-constrained problem, which suggests that existing algorithms used to solve separate chance-constrained versions would be no-longer efficient. We then show that the joint probability functions would also display S-shaped curves, but the non-concave regions may be difficult to be eliminated. We, therefore, develop a new algorithm based on simulation and cut generation to find good staffing solutions. To support the cutting plane method, we develop a *concave-identification* procedure to efficiently identify staffing solutions that belong to the concave regions. We also develop a local-search procedure to further improve a solution returned by the cutting plane method. Our approach allows finding good staffing solutions satisfying the joint chance constraints through simulation and linear programming. We also provide numerical results using call center examples of

up to 89 call types and 65 agent groups to show the advantages of our proposed joint chance-constrained model and solution algorithm.

The rest of this paper is structured as follows. We present the formulation of the joint chance-constrained problem in Sect. 2. In Sect. 3, we present our main methodology to solve the problem, including a sample average approximation (SAA) formulation of the joint chance-constrained problem and an investigation of the shapes of the joint probability functions, and details about the cutting plane method. We then present our numerical results in Sect. 4, and finally, Sect. 5 concludes.

2 Joint chance-constrained staffing optimization

In a staffing optimization problem, we aim at minimizing the total cost of agents while satisfying some constraints on the performance measures, i.e., QoS. We consider a multi-skill call center in which many call types are handled by different agent groups. There are K call types, indexed from 1 to K, I agent groups, numbered from 1 to I. For notational convenience, for any integer number $N \in \mathbb{N}$, we denote the set $\{1, \ldots, N\}$ by $[N]$. The separate chance-constrained (CC) staffing optimization problem (Ta et al. 2020; Chan et al. 2016) is given as

$$\textbf{(P1)} \quad \begin{cases} \underset{x}{\text{minimize}} & c^{\mathrm{T}}x = \sum_{i=1}^{I} c_i x_i \\ \text{subject to} & \mathcal{F}_k(x) = \mathbb{P}[S_k(x) \geq s_k] \geq 1 - \delta_k, \quad k = 1, \ldots, K \\ & \mathbb{P}[S_0(x) \geq s_0] \geq 1 - \delta_0 \\ & x \geq 0 \text{ and integer} \end{cases} \tag{1}$$

where $x = (x_1, \ldots, x_I)$ is a staffing vector representing the number of agents in each group, $c = (c_1, \ldots, c_I)$ is a cost vector, $S_k(x)$ is the SL of call type k, $\mathcal{F}_k(x)$ is the separate chance-constrained function of call type k, S_0 is the aggregated SL, s_k is the SL target of call type k, s_0 is the target for the aggregated SL and δ_k, δ_0 is the risk level selected by the managers. Accordingly, the joint CC version requires a target level for the probability that all the SL requirements are satisfied simultaneously.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^{\mathrm{T}}x = \sum_{i=1}^{I} c_i x_i \\ \text{subject to} \quad & \mathbb{P}\begin{bmatrix} S_1(x) \geq s_1 \\ S_2(x) \geq s_2 \\ \cdots \\ S_K(x) \geq s_K \end{bmatrix} \geq 1 - \delta \\ & \mathbb{P}[S_0(\mathbf{x}) \geq s_0] \geq 1 - \delta_0 \\ & \mathbf{x} \geq 0 \text{ and integer,} \end{aligned}$$

or a compact version

$$\textbf{(P2)} \quad \begin{cases} \underset{x}{\text{minimize}} & c^{\mathrm{T}}x = \sum_{i=1}^{I} c_i x_i \\ \text{subject to} & \mathcal{F}(x) = \mathbb{P}\left[\min_k \{S_k(x) - s_k\} \geq 0\right] \geq 1 - \delta \\ & \mathbb{P}[S_0(x) \geq s_0] \geq 1 - \delta_0 \\ & x \geq 0 \text{ and integer.} \end{cases}$$

where $\mathcal{F}(x)$ is the joint chance constrained function.

As we can see, both models have constraints for the aggregated service level. In the separate CC formulation, each call type is considered independently but in the joint CC version (**P2**), all individual requirements are combined into a single constraint. The joint CC formulation has the advantage of being able to account for the relationship between SLs from different call types. However, this advantage also makes the joint CC problem more difficult to handle (Ahmed and Shapiro 2008).

Before exploring a method to solve (**P2**), we show some properties of the joint CC problem (**P2**) in the property below, which indicates that the joint CC formulation is more conservative but would be robust than the separate CC versions (**P1**), in the sense that a solution to the separate CC problem would give a low probability that all the SL requirements are meting simultaneously as mentioned in part (i) of Theorem 1 of Xie et al. (2017). More precisely, any feasible solution of the joint CC problem (**P2**) is also feasible to the separate CC problem (**P1**). On the other hand, if a solution x is feasible to the separate CC problem (**P1**), then the probability value in the joint chance constraint can be bounded as

$$\min_k \left\{\mathbb{P}\left[S_k(x) \geq s_k\right]\right\} \geq \mathbb{P}\left[\min_k \{S_k(x) - s_k\} \geq 0\right] \geq 1 - K\delta. \tag{2}$$

In our context, these above statements can be easily proved as follows. Let assume that a staffing solution x satisfies the separate constraints, we have the chain

$$\begin{aligned} 1 - \mathbb{P}\left[\min_k \{S_k(x) - s_k\} \geq 0\right] &= \mathbb{P}\left[\exists k \in [K] \middle| \{S_k(x) - s_k\} < 0\right] \\ &\leq \sum_{k=1}^{K} \mathbb{P}\left[\{S_k(x) - s_k\} < 0\right] \\ &= \sum_{k=1}^{K} \left(1 - \mathbb{P}\left[\{S_k(x) - s_k\} \geq 0\right]\right) \\ &\leq K\delta, \end{aligned}$$

where the last inequality is due to $\mathbb{P}[\{S_k(x) - s_k\} \geq 0] \geq 1 - \delta$ for all call type $k \in [K]$. So, we have inequality $1 - \mathbb{P}[\min_k\{S_k(x) - s_k\} \geq 0] \leq K\delta$, which is also the desired result.

Inequality (2) tells us that the joint CC version is more conservative than the separate CC version, in the sense that all solutions being feasible to the joint CC problem are also feasible to the separate CC counterpart. It also says that, in theory, a solution to the separate CC problem would give a low joint probability value, especially when the number of call types becomes large. This is an important property of the joint CC problem, which we will illustrate in more detail by the following simple example. We use a small call center example with 2 call types and 2 agent groups and we refer the reader to the experimental section for a detailed description of the example. The target of the SL for each call type is $s_k = 0.8$. We set risk targets as $\delta = \delta_k = 0.4$ for all $k \in [K]$. We chose a staffing vector $x = (12, 16)$ and use simulation to estimate the probability values. We see that, $\mathbb{P}[S_1 \geq s_1] = 0.665$ for call type 1 and $\mathbb{P}[S_1 \geq s_1] = 0.6$ for call type 2. However, the joint probability of the two call type requirements is $\mathbb{P}[S_1 \geq s_1 \ \& \ S_2 \geq s_2] = 0.55$, which is lower than the target $1 - \delta = 0.6$. Moreover, for some instances of the extra-large call center example considered in Sect. 4, we observe that the joint probability value could be less than 0.1 even all the individual probability values are above 0.8.

It is also interesting to look at the differences between the joint probability function $\mathcal{F}(x)$ and separate probability functions $\mathcal{F}_k(x)$ as well as expected SLs $\overline{S}_k$. We have the following proposition.

Proposition 1 *For any call type $k \in [K]$ served by group $i \in [I]$, we have $\lim_{x_i \to \infty} \mathcal{F}_k(x) = 1$ and $\lim_{x_i \to \infty} \overline{S}_k(x) = 1$. This is however not the case for the joint probability function, i.e., for any group $i \in [I]$ and staffing solution x, if group i does not serve all call types and $\mathcal{F}(x) < 1$ then $\mathcal{F}(x)$ may be bounded from above by a constant that is less than one.*

Proof Clearly, if group i can serve call type k, then if we increase the number of agents in the group to infinity, we will have enough agents to serve all customers instantly, so the corresponding long-term expected SL value should approach one, and the probability value $\mathbb{P}[S_k(x) \geq s_k]$ approaches one as well. On the other hand, for the joint probability function, assume that there is a call type h that cannot be served by group i, then adding agents to group i would not affect (or the influence is very small) S_h, then would not affect $\mathbb{P}[S_h(x) \geq s_h]$. Since we have $\mathbb{P}[S_h(x) \geq s_h] \geq \mathcal{F}(x)$, i.e., the value of $\mathcal{F}(x)$ is bounded by $\mathbb{P}[S_h(x) \geq s_h]$, which could be less than 1 when $x_i \to \infty$. Note that for most of the realistic call center systems that we are aware of, there is no group that is able to serve all call types. □

In general, the joint probability function has some properties distinguishing it with the separate chance-constraints and expected SL constraints, raising a question that whether standard methods used to solve the separate CC and expected SL problems (Atlason et al. 2008; Ta et al. 2020) still apply. We discuss this in the next section, where we show that the joint probability function still displays a S-shaped curve, but identifying the concave region of the function is more challenging, thus requires new investigations.

3 Solution method

In this section, we describe a simulation-based method to solve the joint CC staffing optimization problem. Similar to previous studies (Atlason et al. 2008; Çezik and L'Ecuyer 2008; Ta et al. 2020), to make the problem practical, we use simulation to approximate the joint probability function. We then show that the joint probability function also has an S-shaped curve, suggesting that the standard cutting plane method (Atlason et al. 2008; Çezik and L'Ecuyer 2008) would be still useful. This method only works if all the solutions used to generate linear cuts belong to the concave region of the joint probability function. We, therefore, develop a heuristic approach to efficiently identify the concave region of the joint probability function.

3.1 Sample average approximation

In multi-skill call centers, it is not possible to exactly compute the joint probability functions, so we approximate the joint CC problem (**P2**) using the sample average approximation (SAA) approach and solve the SAA problem instead. More precisely, using simulation, we generate N independent scenarios of the random SL to obtain an estimate of the joint probability function $\mathcal{F}(x)$. The SAA version of (**P2**) can be formulated as

$$
\text{(P3)} \quad \begin{cases} \underset{x}{\text{minimize}} & c^{\mathrm{T}}x = \displaystyle\sum_{i=1}^{I} c_i x_i \\ \text{subject to} & \widehat{\mathcal{F}}_N(x) = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} \mathbb{I}\left[\min_k \{\widehat{S}_k^n(x) - s_k\} \geq 0\right] \geq 1-\delta \\ & \widehat{\mathcal{F}}_N^0(x) = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} \mathbb{I}[\widehat{S}_0^n(x) \geq s_0] \geq 1-\delta_0 \\ & x \geq 0 \text{ and integer} \end{cases} \tag{3}
$$

where $\widehat{S}_k^n(x)$ is a realization of the service level of call type k, for the n-th simulated scenario, under a staffing vector x. The SAA problem is an integer program with nonlinear constraints. As shown in Ta et al. (2020), a solution found by the SAA version will converge to an optimal solution of the original problem when the samples size N goes to infinity. In other words, one can find a good solution by solving the SAA version using appropriate sample sizes.

Since we always need to solve SAA problems to find solutions, we investigate the relation between the SAA version of the joint CC and separate CC problems. To facilitate our exposition, let us first consider the SAA counterpart of the separate CC problem (**P1**) as

$$\textbf{(P4)} \quad \begin{cases} \underset{x}{\text{minimize}} & c^{\mathrm{T}}x = \sum_{i=1}^{I} c_i x_i \\ \text{subject to} & \widehat{\mathcal{F}}_N^k(x) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\widehat{S}_k^n(x) \geq s_k\right] \geq 1-\delta, \quad k = 1, \ldots, K \\ & \widehat{\mathcal{F}}_N^0(x) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[\widehat{S}_0^n(x) \geq s_0] \geq 1-\delta_0 \\ & x \geq 0 \text{ and integer} \end{cases}$$

The following proposition shows the probability that an SAA solution to the joint CC problem is also feasible to the SAA problem of the separate CC problem.

Proposition 2 *If the joint probability function in (**P3**) and the separate probability functions in (**P4**) are approximated by the same set of realizations* $\{\widehat{S}_k^n(x) | \, n \in [N], k \in [K]\}$*, then any feasible solution of (**P4**) is also feasible to (**P3**). Otherwise, if they are using independent sets of realizations, then for any solution* x *being feasible to the SAA problem (**P3**), for any* $k \in [K]$ *and any* $\epsilon > 0$*, we have*

$$\mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) \geq 1-\delta-\epsilon\right] \geq 1 - \exp\left(\frac{-2N\epsilon^2}{9}\right).$$

Proof The first claim is obvious. For the second claim, since x is feasible to (**P3**), we have $\widehat{\mathcal{F}}_N(x) \geq 1-\delta$. Thus,

$$\begin{aligned} &\mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) \geq 1-\delta-\epsilon\right] \\ &\geq \mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) - \widehat{\mathcal{F}}_N(x) \geq -\epsilon\right] \\ &= \mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) - \mathcal{F}^k(x) + \mathcal{F}^k(x) - \mathcal{F}(x) + \mathcal{F}(x) - \widehat{\mathcal{F}}_N(x) \geq -\epsilon\right] \\ &\geq \mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) - \mathcal{F}^k(x) - (\widehat{\mathcal{F}}_N(x) - \mathcal{F}(x)) \geq -\epsilon\right], \end{aligned} \tag{4}$$

where the last inequality is due to the fact that $\mathcal{F}^k(x) \geq \mathcal{F}(x)$. Now we define a random variable $Z = \mathbb{I}[S_k(x) \geq s_k] - \mathbb{I}\left[\min_k\{S'_k(x) - s_k\} \geq 0\right] + \mathcal{F}(x) - \mathcal{F}^k(x)$. Here we assume that the random SL $S_k(x)$ in the first indicator function and $S'_k(x)$ in the second indicator are independent. We see that $\widehat{\mathcal{F}}_N^k(x) - \mathcal{F}^k(x) - (\widehat{\mathcal{F}}_N(x) - \mathcal{F}(x))$ is a sample approximation of $\mathbb{E}[Z]$ and $\mathbb{E}[Z] = 0$ and $Z \in [-2, 1]$. Thus, using the Hoeffding's inequality (Hoeffding 1994) we have

$$\mathbb{P}\left[\widehat{\mathcal{F}}_N^k(x) - \mathcal{F}^k(x) - (\widehat{\mathcal{F}}_N(x) - \mathcal{F}(x)) < -\epsilon\right] \leq \exp\left(\frac{-2N\epsilon^2}{9}\right). \tag{5}$$

Combining (4) and (5) we obtain the desired inequality. □

Proposition 2 implies that the probability that an SAA solution of the joint CC problem (**P3**) is ϵ-feasible to the separate CC problem (**P4**) converges to one exponentially fast when the sample size N goes to infinity, for any $\epsilon > 0$. So if we select an arbitrarily small ϵ, then with a sample size N sufficiently large, an SAA solution of the (**P3**) will be "almost" feasible to the separate CC problem (**P4**). Moreover, as a result of Proposition 2, the probability that a solution give by (**P3**) is ϵ-feasible to (**P4**) is greater than $1 - \rho$, for any $\rho > 0$, if we select $N \geq -9\ln(\rho)/(2\epsilon^2)$. For example, if we use $N > 8300$ samples, then with a probability of 0.95, any solution to (**P3**) is 0.99-feasible to (**P4**).

3.2 Concavity of the joint probability function

The main challenge when solving the staffing optimization problem in multiskill call centers is the nonlinearity of the probability functions. Previous studies (Çezik and L'Ecuyer 2008; Atlason et al. 2008; Avramidis et al. 2010; Ta et al. 2020) make use of the cutting plane method, which is based on the observation that the probability or expected SL functions display S-shaped curves and optimal solutions typically lie in the concave parts. Note that in practice, the management would always wants the risk level to be at a low value (e.g, $\delta \leq 0.3$) so that optimal solutions always lie in the concave part. In our context, we also investigate the use of the cutting plane method due to its practical viability. Nevertheless, as shown in the previous section, the joint CC problem has some properties that make it different from the separate CC and expected long-term SL versions studied in previous studies. So, it is crucial to verify the S-shaped property in the context. We do it in the following.

We first take the example of a medium call center with 6 call types and 8 agent groups considered in Sect. 4 and we refer the reader to the section for a detailed description. Figure 1 shows how the joint probability function $\widehat{\mathcal{F}}_N(x)$ varies when we increase the number of agents in group 4. Starting from an initial staffing vector $x^0 = (18, 21, 16, 18, 20, 18, 17, 19)$, we keep adding agents to x_4 from 18 to 69. As we can see, the joint probability value increases from 0 to 1, and the probability function displays an "S-shaped" form. We also try to vary the numbers of agents in two groups. In Fig. 2 we plot a 3D graph of the joint probability function as a function of the numbers of agents in Groups 4 and 7. For this 3D graph, we start from initial staffing $x^0 = (18, 20, 19, 18, 17, 20, 21, 19)$ and increase the number of agents in the fourth group x_4 from 18 to 58 and the seventh group x_7 from 21 to 61. When the number of agents in two groups increases, the probability value increases very quickly from 0 to 1. We also see that the joint probability also displays an "S-shaped" curve.

However, when investigating the shape of the joint probability function numerically, we also observe that the joint probability function does not always vary from zero to one and has the "S-shape" form. It happens sometimes that when we increase the number of agents in a group that does not serve all call types, the joint probability value increases very slowly, or there is almost no increase. In Fig. 3, we also take the example of 6 call types and 8 agent groups with initial staffing $x^0 = (18, 21, 16, 18, 20, 18, 17, 19)$ and plot the values of $\widehat{\mathcal{F}}_N(x)$ when x_2 varies from 21 to 61. We see that when the number of agents increases from 21, the joint probability value increases very slowly from

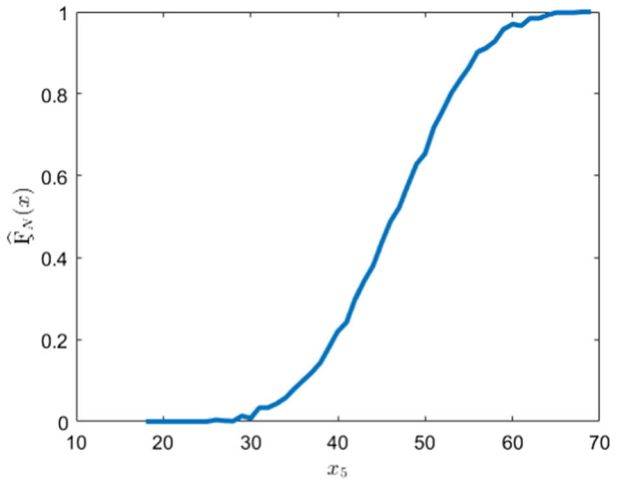

Fig. 1 S-shaped form of the joint probability function

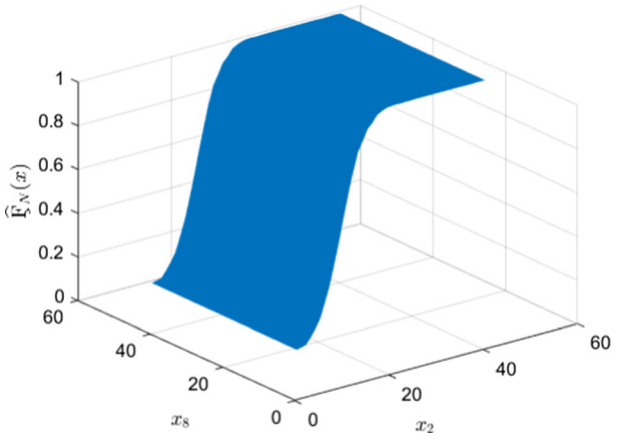

Fig. 2 A 3D surface plot of the joint probability function

zero and then fluctuates around 0.05. The value of 0.05 is very small and that means the joint probability value is almost unchanged, noting that the difference between the probability values at each staffing point is very small so we can consider the zigzag pattern as a result of simulation-noises. Therefore, in this context, to improve the joint probability value, one needs to add more agents groups that really affect the joint probability function. This observation is also consistent with (2) and Observation 1 and drives the design of the *concave-identification* method in the next section.

We also provide a comparison of the joint probability and a separate probability function using the large call center with 20 call types and 15 agent groups (see the experimental section for more details). We start from an initial staffing x^0 = (28, 0, 16, 16, 0, 0, 18, 11, 13, 0, 12, 6, 0, 0, 0) and increase the fifth agent group x_5 from 0 to 53 while keeping the agent numbers in the other groups unchanged. We simulate

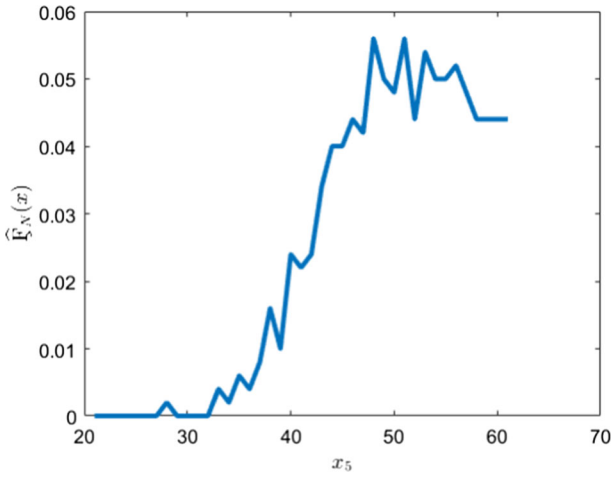

Fig. 3 Non "S-shaped" curve example

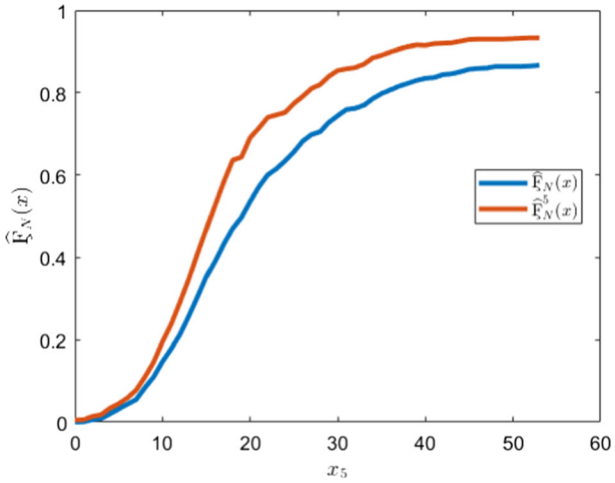

Fig. 4 Comparison between joint and separate probability values

and estimate the value of both joint probability and separate probability functions. As expected, the values given by the joint probability function are always smaller or equal to the minimum value of separate probability functions at each point of x_5, leading to a lower curve.

3.3 Cut generation

In this section, we discuss the cutting plane method in detail. In analogy to previous studies (Atlason et al. 2008; Çezik and L'Ecuyer 2008; Chan et al. 2016), the method is supported by the S-shaped forms of the probability functions and the concavity of

the probability functions in the region of interest, i.e., the region containing feasible staffing solutions. The method is based on an iterative procedure in which at each iteration if the chance constraints are unsatisfied, i.e., there is a probability value smaller than the target, and the current staffing solution lies in the concave region, then we can create linear cuts based on subgradient information to construct outer-approximations of the probability functions. The iterative procedure stops when we find a solution satisfying all the chance constraints.

The cutting plane method requires that all the points used to generate linear cuts need to belong to the concave regions of the probability functions, otherwise the procedure would return bad solutions. To eliminate the non-concave region, previous studies (Çezik and L'Ecuyer 2008; Chan et al. 2016) make use of a fluid scheduling model (Bassamboo et al. 2006). This model has heuristic linear constraints to cover a fraction of α_k of the arrival rate of call type k, as described in Chan et al. (2016). New additional variables $\omega_{k,i} \geq 0$ are required in this model which defines the number of agents of group i handling calls of type k. One can construct some constraints for the staffing problem before adding subgradient cuts using the fluid scheduling model

$$\begin{aligned}
&\sum_{i \in G_k} \mu_{k,i} \omega_{k,i} \geq \alpha_k \Lambda_k && k \in [K] \\
&\sum_{k \in S_i} \omega_{k,i} \leq x_i && i \in [I] \\
&\omega_{k,i} \geq 0 && k \in [K] \quad i \in [I]
\end{aligned}$$

where Λ_k is the arrival rate of call type k, $\{\alpha_k\}$ are some parameters that should be selected to get an initial solution lying in the concave region of $\widehat{\mathcal{F}}_N(x)$, G_k is the set of groups that can handle call type k and S_i is the set of call types that can be handled by group i.

It has been shown that the fluid scheduling model works well for staffing optimization problems under separate chance constraints or long-term expected SL constraints (Çezik and L'Ecuyer 2008; Ta et al. 2020). However, it is not the case in our context. More precisely, we observe that it is not easy to select good parameters α_k to identify the concave regions of $\widehat{\mathcal{F}}_N(x)$. This model often results in too low staffing solutions, i.e., fails to eliminate the non-concave part, or results in too high staffing solutions, i.e., it eliminates good staffing solutions. This issue will be seen more clearly in our experimental section. Thus, we propose a heuristic phase to overcome this issue. The idea is that we start from low values of α_k, $k \in [K]$, to make sure that we do not eliminate good solutions. We then iteratively use simulation to compute probability values, and if there is a too small joint probability value, i.e, less than a given threshold, then we add more agents to improve the joint probability value. In the experiments, we will show that this method is really beneficial in tackling the aforementioned issue of the fluid model.

In the following, we describe in detail how to create linear cuts based on the concept of subgradient. First, we consider a probability function $f(x)$. Let x^* be the current solution and $q(x^*)$ be a subgradient vector of $f(x)$ at point x^*. This subgradient vector can be estimated numerically as follows. For each given x^*, we generate I staffing vectors $x_1^*, \ldots, x_I^*$ where $x_i^* = x^* + de_i$ where e_i is a vector of size I with a value

of 1 at the i-th position and 0 elsewhere, and $d \in \mathbb{N}_+$ is a step size. Then, we use simulation to compute $f(x)$ and $f(x_i^*)$, $i \in [I]$ and estimate the i-th element $q_i(x^*)$ as follows.

$$q_i(x^*) = [f(x_i^*) - f(x^*)]/d. \tag{6}$$

As in previous papers (Chan et al. 2016; Ta et al. 2019), d is usually set to 1, and sometimes d may be increased to 2 or 3 to deal with simulation noises (e.g., the number of scenarios N is small) or the subgradient is not computed as expected, e.g., $q_i(x^*) < 0$ for some i. In our context, to better understand how the value of d affects the performance of our algorithm, we will test with $d = 1$, 2, or 3. Details can be found in the experimental section.

Now, suppose that we have $q(x^*)$ as an estimation of the subgradient vector of function f at point x^*. From the assumption that x^* belongs to the concave region of f and the properties of a subgradient vector, we obtain the following valid inequality $f(x^*) + q(x^*)(x - x^*) \geq f(x)$. We want to find x such that $f(x) \geq 1 - \delta$, leading to the following valid inequality

$$q(x^*)^{\mathrm{T}} x \geq 1 - \delta - f(x^*) + q(x^*)^{\mathrm{T}} x^*, \tag{7}$$

which is linear. These linear cuts can be added after each iteration to obtain linear programs of the form

$$\min_{\substack{x \in \mathbb{N}^I \\ \omega \in \mathbb{R}_+^{K \times I}}} \left\{ c^T x \quad | \quad Ax \leq b, \quad Hx + K\omega \leq h \right\}, \tag{8}$$

where $Ax \leq b$ refers to the set of subgradient cuts and $Hx + K\omega \leq h$ are constraints given by the fluid model. The problem in (8) is a mixed-integer linear program and can be solved conveniently by a general-purpose solver, e.g., CPLEX or Gurobi. Furthermore, one can show that the cutting plane method will return an optimal solution to the SAA problem if all the assumptions perfectly hold, i.e., all the points used to generate cuts belong to the concave regions of the probability functions, and all sub-gradients $q(x^*)$ are well estimated, in the sense that $f(x^*) + q(x^**)(x - x^*) \geq f(x)$ for any x in the concave region of $f(\cdot)$. These assumptions, however, may not hold in practice, leading to sub-optimal solutions. Hence, in the next section, we develop a local-search procedure to partially address the issue.

3.4 Simulation-based algorithm

In this section, we describe our algorithm in detail. Algorithm 1 describes the detailed steps of our algorithm. First, we need to choose a threshold $\rho \in [0, 1]$ to identify the concave regions of the probability functions, a step size d, parameters $\alpha_k, k \in [K]$ for the fluid scheduling model. Then, we construct some preliminary constraints using the fluid scheduling model and add them to the mixed-integer linear program (MILP) in (8). Solving this problem by a MILP solver gives us an initial staffing solution.

Algorithm 1: Simulation-based cutting plane for the joint CC staffing problem

 # Set up some parameters and preliminary constraints
Select a threshold ρ to define "concave regions", a step size d. Set a limit value for the step size $d_{\max}$, choose α_k and add preliminary constraints using the fluid scheduling model to the linear model (8).
 # Simulation-based cutting plane
Solve the linear model (8) to get an initial staffing x^0.
Simulate with x^0 to get probability values and set $x = x^0$.
while $\widehat{\mathcal{F}}_N(x) < 1-\delta$ *or* $\widehat{\mathcal{F}}^0_N(x) < 1-\delta_0$ **do**
  # Identify points in the concave regions
  $\bar{x} \leftarrow$ *Concave-Identification*(x). *# Algorithm 2*
  for *each probability function* $f(\cdot)$ *that does not satisfy the chance constraints* **do**
    Estimate a subgradient vector $q(\bar{x})$ using simulation and (6).
    Add linear cut $q(\bar{x})^{\mathrm{T}} x \geq 1-\delta-f(\bar{x})+q(\bar{x})^{\mathrm{T}}\bar{x}$ to the linear programming model (8).
  Solving linear model (8) to get a new staffing candidate x.
  Simulate to get new probability values.
 # Local search to further improve the solution
$x \leftarrow$ *Local-search*(x) *# Algorithm 3*
Return x.

As being said, it is crucial in our algorithm to add cuts based on points that are in the concave regions. To do so, we need to check whether the probability values are large enough. If there is any probability value less than the chosen threshold ρ, we need to add more agents to improve the joint probability value. In the cases of separate CC or expected SLs, it can be done simply by adding more agents to groups that serve unsatisfied call types. For our joint CC problem, it is not straightforward to choose groups that we should add agents to, in order to improve the joint probability values. To this end, we look closely at the joint probability function and write

$$\mathbb{P}\left[\min_k\{S_k(x)-s_k\}\geq 0\right]=\sum_{h\in[K]}\mathbb{P}\left[h=\operatorname{argmin}_{k\in[K]}\{S_k(x)-s_k\}\ \&\ S_h(x)-s_h\geq 0\right].$$

So, to effectively improve $\mathbb{P}\left[\min_k\{S_k(x)-s_k\}\geq 0\right]$, we select

$$\overline{h}=\operatorname{argmax}_{h\in[K]}\mathbb{P}\left[h=\operatorname{argmin}_{k\in[K]}\{S_k(x)-s_k\}\ \&\ S_h(x)-s_h\geq 0\right].$$

In other words, we select a call type that affects the most the joint probability value. Note that, in the context of the SAA method, $\overline{h}$ can be selected as follows.

$$\overline{h}=\operatorname{argmax}_{h\in[K]}\left\{\sum_{n=1}^{N}\mathbb{I}\left[h=\operatorname{argmin}_{k\in[K]}\{\widehat{S}^n_k(x)-s_k\}\ \&\ \widehat{S}^n_h(x)-s_h\geq 0\right]\right\}. \quad (9)$$

So, the idea is that we keep selecting $\overline{h}$ and increasing the number of agents and simulating the probability functions until there is no probability value that is less than threshold ρ. We describe this procedure in Algorithm 2, where s is a step size. Note that s can be chosen adaptively according to the joint probability values, e.g., we can choose a large s when the joint probability value is small, and small s when the joint

probability value is close to ρ. We discuss different ways to select s in our experimental section.

Algorithm 2: *Concave-Identification* (x)

Input: Staffing vector x
Output: Staffing vector $\bar{x}$ lying in the concave region of $\widehat{\mathcal{F}}_N(\cdot)$.
while $\widehat{\mathcal{F}}_N(x) \leq \rho$ *or* $\widehat{\mathcal{F}}^0_N(x) \leq \rho$ **do**
  # Select the call type that affects the most the joint probability function.
  Compute $g_h = \sum_{n=1}^{N} \mathbb{I}\left[h = \text{argmin}_{k\in[K]}\{\widehat{S}^n_k(x) - s_k\} \;\&\; \widehat{S}^n_h(x) - s_h \geq 0\right]$ for all $h \in [K]$.
  Select $\overline{h} = \text{argmax}_h\{g_h\}$.
  # Add more agents to improve $\widehat{\mathcal{F}}_N(x)$
  Select a group i that serves call type $\overline{h}$.
  $x_i \leftarrow x_i + s$.
  Simulate with the new staffing x to obtain new $\widehat{\mathcal{F}}_N(x)$, $\widehat{\mathcal{F}}^0_N(x)$.
end
Return x.

Algorithm 3: *Local-search* (x)

Input: Staffing vector x
Output: Staffing vector $\bar{x}$ that yields a better cost while being feasible to the constraints
while $\widehat{\mathcal{F}}_N(x) \geq 1 - \delta$ & $\widehat{\mathcal{F}}^0_N(x) \geq 1 - \delta_0$ **do**
  # Select the call type that affects the least the joint probability function.
  Compute $g_h = \sum_{n=1}^{N} \mathbb{I}\left[h = \text{argmin}_{k\in[K]}\{\widehat{S}^n_k(x) - s_k\} \;\&\; \widehat{S}^n_h(x) - s_h \geq 0\right]$ for all $h \in [K]$.
  Select $\overline{h} = \text{argmin}_h\{g_h\}$.
  # Remove agents
  Select a group i that serves call type $\overline{h}$
  $x_i = x_i - s$
  Simulate with the new staffing vector x.
end
Return the best staffing solution found.

In the main part of the algorithm, we use the cutting plane method and simulation to find a good solution to the SAA problem. We keep adding linear cuts for violated chance constraints and solve the linear program (8) until we find a feasible solution. At each iteration, we use simulation to verify whether the current solution belongs to the concave regions by comparing the probability values to the chosen threshold ρ. If it is not the case, we use Algorithm 2 to shift the current solution to a new one that yields probability values above ρ. Then, for each unsatisfied chance constraint, we estimate the subgradient vector by (6). As mentioned in previous work (Çezik and L'Ecuyer 2008; Ta et al. 2020), even under the right conditions, subgradient estimates may have non-positive elements, which could lead to bad cuts. An additional step is required to check the quality of such subgradient estimates carefully. If the subgradient vectors satisfy some basic requirements, i.e., having non-negative entries, we create linear cuts by (7) and add them to the linear program (8). We then solve this program with a general-purpose solver (e.g. CPLEX) to get a new solution candidate. Using simulation, we check if that solution is feasible to the chance constraints. If it is the case, we quit the loop and return the best solution found. Otherwise, we move to the next iteration with the new solution candidate.

The last step of Algorithm 1 involves a local search procedure to further improve a solution returned by the cutting plane method. The idea is to see whether one can remove agents while keeping all the constraints satisfied. In analogy to the *concave-identification* procedure, we identify a call type $\bar{h}$ that affect *the least* the joint probability value as

$$\bar{h} = \text{argmin}_{h\in[K]} \left\{ \sum_{n=1}^{N} \mathbb{I}\left[h = \text{argmin}_{k\in[K]}\{\widehat{S}_k^n(x) - s_k\} \;\&\; \widehat{S}_h^n(x) - s_h \geq 0\right] \right\}. \tag{10}$$

Then, we can choose a group that serves call type $\bar{h}$ and has the largest cost, and remove agents from that group. This can be done iteratively until we no-longer can remove agents while keeping all the chance constraints satisfied.

4 Numerical experiments

In this section, we provide experimental results based on three call center examples (medium, large, and real-life extra-large call centers). These examples have been used in previous studies (Çezik and L'Ecuyer 2008; Chan et al. 2014, 2016; Ta et al. 2019). The objective is to numerically illustrate the advantage of the joint CC formulation, as compared to its separate CC counterpart, and evaluate the performance of our algorithm.

4.1 Experimental settings

We test our algorithm using targets: $1 - \delta = 0.8$ (i.e. 80%) and $1 - \delta_0 = 0.85$ (i.e. 85%) and three step sizes $d \in \{1, 2, 3\}$. We denote the set of targets as (80%, 85%). We set the agent costs based on the number of skills in the agent's skill set S_i (the number of call types that agent can serve) as $c_i = 1 + 0.1(|S_i| - 1)$, where $|S_i|$ is the cardinality of S_i and $c = (c_1, \ldots, c_i)^T$.

We choose a sample size $N = 1000$ for all the examples. Previous studies have shown that these sample sizes are sufficient to provide good approximations to the probability/expected SL functions (Chan et al. 2014; Ta et al. 2019). We vary the arrival rate, leading to 10 instances for each example. Each instance corresponds to an arrival-rate scenario in the two-stage staffing optimization model considered in Ta et al. (2019) and we refer the reader to this paper for more details. For each instance, we solve the joint CC problem by Algorithm 1 and the separate CC counterpart by the classical algorithms in (Çezik and L'Ecuyer 2008; Ta et al. 2019). For each solution obtained from the separate CC problem, we compute resulting joint probability values. The idea is to show how such solutions perform in the context of joint chance constraints.

In the following, we describe in detail our heuristic method used to identify staffing solutions that belong to the concave regions, i.e., Algorithm 2. As mentioned, this method is important, motivated by the fact that adjusting the parameters α_k in the fluid scheduling model is neither easy nor efficient, even manually, especially for large-size examples with many call types and agent groups. To effectively perform this task, we

consider three heuristic approaches to increase the number of agents in the group $\bar{h}$ identified in (10):

- **Method #1**: We increase the number of agents in the chosen group by 1.
- **Method #2**: We consider two cases
  - If $\widehat{\mathcal{F}}_N(x) < 0.2$ or $\widehat{\mathcal{F}}^0_N(x) < 0.2$, increase the number of agents in the chosen group by 2.
  - Otherwise, increase the number of agents in the chosen group by 1.
- **Method #3**: We consider 4 levels of the probability functions
  - If $\min\{\widehat{\mathcal{F}}_N(x), \widehat{\mathcal{F}}^0_N(x)\} < 0.1$, increase the number of agents by 4.
  - If $\min\{\widehat{\mathcal{F}}_N(x), \widehat{\mathcal{F}}^0_N(x)\} \in [0.1, 0.2]$, increase the number of agents by 3.
  - If $\min\{\widehat{\mathcal{F}}_N(x), \widehat{\mathcal{F}}^0_N(x)\} \in [0.2, 0.3]$, increase the number of agents by 2.
  - Otherwise, increase the number of agents by 1.

For each data set, we run Algorithm 1 with the three heuristic methods above and choose the best method to provide comparisons of the joint CC and separate CC formulations. To evaluate the performance of our *concave-identification* method, we run Algorithm 1, with and without the *concave-identification* phase, and compare the solutions obtained.

The experiments are done using a personal computer with Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz processor and 8192 MB of RAM. We use MATLAB 2017 to implement, run and link to IBM ILOG CPLEX 12.10 to solve mixed-integer linear programs with a time budget of 2 min. The simulation is performed using the Contact-Centers simulation library (Buist and L'Ecuyer 2005) developed under the SSJ simulation package (L'Ecuyer et al. 2003).

4.2 Medium example

First, we test our approach using a medium-size call center of 6 call types and 8 agent groups. We use the same settings as in previous studies (Chan et al. 2014, 2016; Ta et al. 2019), i.e., (i) the callers do not abandon immediately in case they have to wait, (ii) patience times follow an exponential distribution with means between 36 and 52 min, and (iii) all service times follow Log-Normal distributions with means between 5.1 and 11.3 min. The acceptable waiting times are chosen as $\tau_k = \tau_0 = 120$ (seconds) and the target for SLs are $s_k = s_0 = 80\%$. We choose step size $d = 2$ for all test cases. We choose the fluid parameters $\alpha = (1, 4, 1.2, 1, 3)$ and choose **Method #2** for identifying the concave regions, as it performs the best among those proposed, in terms of CPU time and solution quality.

We show comparison results for the 10 instances in Table 1. As we can see, in terms of agent cost and running time, the separate CC model performs better than the joint CC model, in the sense that the cutting plane algorithm returns better costs for the separate CC than for the joint CC model in 9/10 instances and has better running time in 9/10 instances. This is not surprising because, as shown in Sect. 2, the joint CC formulation is more conservative. However, the percentage gaps between the final costs given by the joint CC and separate CC models are quite low (less than 1%).

Table 1 CPU times, probability values, agent costs for the medium call center example

Instance	Joint CC solutions				Separate CC solutions				Gap (%)
	Time (s)	Joint prob. values		Cost	Time (s)	Joint prob. values		Cost	
		$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$		
1	985	0.81	0.87	196.7	777	0.77	0.85	195.6	0.56
2	1313	0.81	0.86	190.4	742	0.79	0.86	190.1	0.16
3	863	0.83	0.87	189.6	473	0.72	0.86	188.8	0.42
4	1450	0.81	0.86	196.5	752	0.76	0.85	196.5	0.00
5	805	0.81	0.87	182.9	676	0.76	0.86	182.1	0.44
6	862	0.81	0.88	190.2	951	0.80	0.86	189.7	0.26
7	1027	0.80	0.87	175.3	220	0.73	0.85	174.0	0.74
8	978	0.80	0.86	160.7	520	0.77	0.85	159.5	0.75
9	924	0.82	0.87	193.5	702	0.78	0.85	192.7	0.41
10	1356	0.80	0.85	210.0	618	0.75	0.85	209.5	0.38

Table 2 Comparison results for the cutting plane method without using the *concave-identification* procedure

Set of $\{\alpha_k\}$	Instance	Time	Prob. values		Cost	Gap (%)
			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$		
1	1	175	1.00	1.00	321.4	63.40
	2	156	1.00	1.00	311.3	63.50
	3	149	1.00	1.00	309.8	63.40
2	1	596	1.00	1.00	551.4	180.33
	2	272	0.95	1.00	235.6	23.74
	3	693	1.00	1.00	1150.6	506.86
3	1	849	0.80	0.87	196.8	0.05
	2	1126	0.81	0.86	190.7	0.16
	3	234	0.83	0.97	220.5	16.30

Furthermore, solutions from the joint CC model have better quality, in the sense that solutions given by the separate CC model always give joint probability values that are significantly lower than the target.

We evaluate the efficiency of our *concave-identification* methods as compared to the widely-used fluid scheduling model and report comparison results in Table 2. We use three sets of $\{\alpha_k,\ k \in [K]\}$, from small to large values, for the fluid scheduling model. When we run the algorithm with only the fluid scheduling model, the CPU time is very low because it does not need to perform simulations to identify the concave regions of the probability functions. However, it also gives very high final costs and probability values. The percentage gaps between the costs given by the algorithm with and without our *concave-identification* method vary from 0.05 to 506.86%. This indicates the benefit of using Algorithm 2 to avoid bad solutions.

In Table 3, we also provide numerical results comparing the three implementations of the *concave-identification* method: **Method #1**, **Method #2** and **Method #3** (or

Table 3 Comparison results between three *concave-identification* implementations for the medium example

Instance	CPU time (s)			Cost		
	M.#1	**M.#2**	**M.#3**	**M.#1**	**M.#2**	**M.#3**
1	1034	985	**928**	**196.7**	**196.7**	198.0
2	1295	1313	**1065**	190.4	190.4	**190.0**
3	884	**863**	1024	189.6	189.6	**189.4**
4	1484	**1450**	1717	**196.5**	**196.5**	198.2
5	**786**	805	1008	**182.9**	**182.9**	183.0
6	880	**862**	1156	**190.2**	**190.2**	190.8
7	1040	**1027**	1312	**175.3**	**175.3**	175.0
8	983	978	**952**	160.7	160.7	160.7
9	**903**	924	925	**193.5**	**193.5**	193.5
10	1364	**1356**	1563	**210.3**	**210.3**	210.5

M.#1, M.#2 and **M.#3** for short). We indicate in bold the best running times and the best costs obtained by the three methods. As we can see, in terms of CPU time, **M.#1** has 2/10 instances, **M.#2** has 5/10 instances and **M.#3** has 3/10 instances having the best running times. Therefore, **M.#2** is faster than the other two methods. Moreover, in terms of cost, **M.#1** and **M.#2** give quite similar costs. **M.#1** and **M.#2** have 7/10 instances and **M.#3** has 2/10 instances having the best costs. However, the gaps between costs given by the three methods are relatively small. In general, the second method is better than the other ones.

4.3 Large example

In this section, we report numerical results for a large call center of $K = 20$ call types and $I = 15$ agent groups. Similar to previous studies, we assume that (i) immediate call abandonment has an odd of 0.001, (ii) the patience times follow exponential distributions with means of 10 min for all call types, and (iii) the service times follow exponential distributions with means of 8 min for all call types. The acceptable waiting times are chosen as $\tau_k = \tau_0 = 20$ (seconds) and the target for SLs are $s_k = s_0 = 80\%(0.8)$. We choose step size $d = 2$ for all call types.

In Table 4, we show numerical results for 10 instances with targets (80%, 85%). We use **M.#3** as the *concave-identification* implementation. For this large call center, the separate CC model gives lower agent costs and requires less running times in all 10 instances. We may need up to 1800 s (30 min) to solve a joint CC instance. Also, the percentage gaps between the agent costs returned by the two models are around 5% and they are significantly larger than those reported for the medium call center example. However, solutions from the joint CC model seem to have much better "joint" quality than those given by the separate CC model, in the sense that the joint probability values given by separate CC solutions are just around 0.6, which are remarkably lower than the targets.

In Table 5, we report comparison results obtained by running our algorithm with and without the *concave-identification* step. We test on three sets of α for the fluid

Table 4 CPU times, probability values, agent cost with the second increase method for the large call center

Instance	Joint CC solutions				Separate CC solutions				Gap (%)
	Time	Prob. values		Cost	Time	Prob. values		Cost	
		$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$		
1	3136	0.80	0.95	198.9	234	0.64	0.88	190.1	4.42
2	2211	0.80	0.96	198.7	297	0.64	0.85	190.5	4.13
3	2622	0.80	0.96	188.3	189	0.63	0.88	179.0	4.94
4	1802	0.81	0.95	190.7	169	0.61	0.87	181.2	4.98
5	2581	0.80	0.95	190.2	225	0.64	0.90	183.1	3.73
6	2669	0.80	0.95	202.4	340	0.64	0.86	193.6	4.35
7	1817	0.80	0.95	200.1	235	0.64	0.88	190.8	4.65
8	3014	0.80	0.95	196.9	375	0.64	0.85	186.7	5.18
9	1881	0.80	0.95	195.8	248	0.61	0.86	186.6	4.70
10	1723	0.81	0.95	198.7	143	0.59	0.86	189.2	4.78

Table 5 Comparison results for the cutting plane method without using the *concave-identification* procedure, symbol "–" indicates the algorithm cannot return a feasible solution after reaching the step-size limit $d_{\max}$

Set of $\{\alpha_k\}$	Instance	Time	Joint prob. values		Cost	Gap (%)
			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$		
1	–	–	–	–	–	–
	–	–	–	–	–	–
	–	–	–	–	–	–
2	1	39	0.84	1.00	324.5	63.15
	2	403	0.85	1.00	326.3	64.22
	3	262	0.88	1.00	305.1	62.03
3	1	131	0.81	1.00	229.8	15.54
	2	419	0.80	1.00	237.7	19.63
	3	678	0.82	1.00	221.6	17.68

scheduling model: (1) $\alpha = 1$ for all call types, (2) $\alpha = 2$ for all call types, (3) $\alpha_1 = 0.4, \alpha_2 = 0.6, \alpha_5 = 0.6, \alpha_6 = 0.3, \alpha_7 = 0.5, \alpha_{12} = 0.4, \alpha_{17} = 0.5, \alpha_{19} = 1.2$ and the remaining parameters are set to 1.5. The latter vector was chosen manually to achieve the best performance. We compare the final costs obtained when running the algorithm with our *concave-identification* method and with the fluid scheduling model alone based on the above sets of $\{\alpha_k\}$. For the first set $\{\alpha_k\}$, the algorithm is not able to return any feasible solution. This is due to the fact that the values of α_k are too small, leading to too low initial staffing solutions and all the subgradient estimates are zero. In these cases, the step size reaches the step limit $d_{\max}$ but the staffing solution is still small and the algorithm is not able to compute subgradient vectors as expected, hence, cannot add linear cuts to the linear program. However, if we use the *concave-identification* method, the issue is resolved and the algorithm is always able to give good solutions.

Table 6 Comparison results between three *concave-identification* implementations for the large example

Instance	Time (s)			Cost		
	M.#1	**M.#2**	**M.#3**	**M.#1**	**M.#2**	**M.#3**
1	3520	**2485**	3136	199.9	198.6	**198.9**
2	2716	2695	**2211**	200.0	200.3	**198.7**
3	2456	**2206**	2622	**187.6**	**187.6**	188.3
4	2291	2363	**1802**	**190.5**	191.2	190.7
5	3105	3004	**2581**	190.3	190.3	**190.2**
6	2772	3132	**2669**	**202.4**	202.9	**202.4**
7	1984	1861	**1817**	199.8	**198.9**	200.1
8	3126	3249	**3014**	**195.8**	196.1	196.9
9	2046	1924	**1881**	196.8	196.1	**195.8**
10	1992	2340	**1723**	**197.6**	198.5	198.7

We report comparison results of the three *concave-identification* implementations in Table 6. We indicate in bold the best CPU times and the best costs. In terms of CPU time, **M.#3** performs faster than **M.#1** and **M.#2** when it has 8/10 instances with better running times, and for the other 2 instances, **M.#2** performs the best. This happens because **M.#3** requires a smaller number of simulations. In terms of cost, **M.#1** gives better results in 5/10 instances while **M.#2** returns better results in 2/10 instances and **M.#3** performs better in 5/10 instances. Thus, in general, **M.#3** performs better than the other two methods.

4.4 Extra-large example

In this section, we test on an extra-large call center example of $K = 89$ call types and $I = 65$ agent groups. Similar to previous studies, we also assume that (i) immediate call abandonment does not occur, (ii) the patience times follow exponential distributions with means of 3 min for all call types, and (iii) the service times follow exponential distributions with means varying from 4.32 to 12.79 min. The acceptable waiting times for all call types are $\tau_k = \tau_0 = 20$ seconds and the targets of SLs are $s_k = 50\%(0.5)$ for all $k = 1, \ldots, 89$ and $s_0 = 80\%(0.8)$. This example is also used in Çezik and L'Ecuyer (2008) and can be found at http://www.iro.umontreal.ca/~lecuyer/myftp/ld-example2/. To achieve good performance, we choose step size $d = 3$ and the third *concave-identification* implementation (**M.#3**) and $\alpha_k = 1$ for all call types.

Table 7 reports our results for the extra-large call center. As we can see, there are significant differences between the joint probability values and costs given by the joint CC model and the separate CC model. The CPU times for the joint CC model are over 70,000 s (19.4 h) while the running times for the separate CC model are less than 17,000 s (4.7 h). Moreover, the percentage gaps (between costs given by the two formulations) are large and vary from 9.35 to 11.22%. These gaps are much larger than those from the medium and large examples. However, the joint probability values given by the solutions of the separate CC model are very small, from 0.08 to 0.16, and much smaller than the target of 80%. This indicates a clear advantage of the joint

Table 7 CPU times, probability values, agent cost for the extra-large call center example

Instance	Joint CC				Separate CC				Gap (%)
	Time (s)	Joint prob. values		Cost	Time (s)	Joint prob. values		Cost	
		$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}^0_N(x)$		
1	81,438	0.80	1.00	914.7	9603	0.08	0.90	829.2	9.35
2	82,220	0.80	0.98	893.3	12,119	0.10	0.85	804.4	9.95
3	80,596	0.80	0.99	926.1	16,632	0.09	0.88	822.2	11.22
4	91,645	0.80	0.96	875.0	13,838	0.10	0.90	789.4	9.78
5	71,005	0.81	0.97	923.3	15,077	0.08	0.91	835.9	9.47
6	73,477	0.80	0.98	885.6	11,123	0.05	0.91	798.1	9.88
7	98,055	0.81	0.99	947.3	14,480	0.16	0.91	852.4	10.02
8	72,152	0.80	0.99	888.7	14,756	0.07	0.92	799.3	10.06
9	76,965	0.80	1.00	887.0	15,039	0.09	0.89	801.6	9.63
10	79,959	0.80	0.99	920.0	13,293	0.11	0.89	823.3	10.51

CC model, as compared to the separate CC one. In fact, if the number of call types is large, the probability that the service levels are simultaneously high for all the call types should be low. Thus, as the number of call types I increases, a solution to the separate CC model would give low joint probability values. We have theoretically explored this fact through Inequality (2) in Sect. 2.

In Table 8, we also report comparison results obtained by running our algorithm with and without the *concave-identification* step. We use three sets of α to test: (i) $\alpha = 1$ for all call type, (ii) $\alpha = 2$ for all call types, (iii) $\alpha_1 = 1.3$, $\alpha_6 = 1$, $\alpha_8 = 1.8$, $\alpha_{19} = 1.2$, $\alpha_{20} = 1.7$, $\alpha_{26} = 1.6$, $\alpha_{31} = 1.1$, $\alpha_{45} = 1.6$, $\alpha_{53} = 1.4$ and the remaining parameters are set to 1.5. We report the percentage gaps between the final costs obtained by running our algorithm with the *concave-identification* method (those reported in Table 7) and with only the fluid scheduling model based on the above sets of α. As we can see, for the first set of α, the costs are very high with gaps varying from 131.70 to 402.48%. For the second set of α, the values of α are larger and we can find better solutions. For the final set, we adjust it manually to achieve the best performance. When using the fluid model alone, the algorithm runs very fast on all three sets of α. However, it also returns very high final costs. The percentage gaps are always above 60%. This indicates that the efficiency of the fluid scheduling model depends on the coefficients α chosen. If the number of call types is large, it is typically difficult to choose a good set of α, leading to bad performance of the fluid scheduling model. We also note that all the solutions obtained by solving the joint CC problem satisfy the separate chance-constraints. This is consistent with our theoretical finding in Proposition 2, saying that the probability that a SAA solution of the joint CC problem is "ϵ-feasible" to the separate CC problem converges to one exponentially fast. In other words, if we select a large enough number of samples N, a SAA solution of the joint CC problem will be *almost* feasible to the separate CC counterpart.

In this example, since the experiments are costly to perform, we do not provide comparison results for the three *non-concave identification* implementations as in the medium and large examples.

Table 8 Comparison results for the cutting plane method without using the *concave-identification* step, for the extra-large example

Set of $\{\alpha_k\}$	Instance	Time (s)	Prob. values		Cost	Gap (%)
			$\widehat{\mathcal{F}}_N(x)$	$\widehat{\mathcal{F}}_N^0(x)$		
1	1	49, 077	0.80	0.99	2119.4	131.70
	2	12, 754	0.82	1.00	3813.8	326.93
	3	79, 072	0.80	0.99	4653.5	402.48
2	1	3145	0.83	1.00	1560.1	70.56
	2	2006	0.82	1.00	1518.3	69.97
	3	2436	0.81	1.00	1550.4	67.41
3	1	3321	0.81	1.00	1516.5	65.79
	2	2587	0.84	1.00	1480.0	65.68
	3	3168	0.80	1.00	1507.4	62.77

5 Conclusion

We have studied a joint CC staffing optimization model in multi-skill call centers. The main advantage of the model, as compared to the separate versions considered in previous studies, is that our joint CC model allows to account for the correlation between different call types, thus would be of interest to the manager. We show some properties of the joint CC model that distinguishes it from the separate CC counterpart and long-term expected SL models studied previously, which leads to the issue that standard solution algorithms may not work well under the joint CC formulation. We have shown that the joint probability function still displays an "S-shaped" curve, but the concave region might be difficult to well identify. We thus proposed a new algorithm based on simulation and linear programming to find a good solution. We also developed a *concave-identification* to help shift a staffing solution to the concave region, and a local-search procedure to further improve a solution returned by the cutting plane method. Our numerical results based on three call center examples of up to 89 call types and 65 agent groups clearly showed the benefits of the joint CC model over the separate CC counterpart. Furthermore, we showed the advantage of our *concave-identification* methods as compared to the standard approaches. In future works, we are interested in staffing optimization models accounting for uncertainty issues occurring in the modeling of call centers, e.g. arrival rate or service rate uncertainty.

References

Ahmed S, Shapiro A (2008) Solving chance-constrained stochastic programs via sampling and integer programming. In: State-of-the-art decision-making tools in the information-intensive age. Informs, pp 261–269

Atlason J, Epelman M, Henderson S (2004) Call center staffing with simulation and cutting plane methods. Ann Oper Res 127:333–358

Atlason J, Epelman M, Henderson S (2008) Optimizing call center staffing using simulation and analytic center cutting-plane methods. Manag Sci 54:295–309

Avramidis A, Chan W, Gendreau M, L'Ecuyer P, Pisacane O (2010) Optimizing daily agent scheduling in a multiskill call center. Eur J Oper Res 200:822–832
Bassamboo A, Harrison J, Zeevi A (2006) Design and control of a large call center: asymptotic analysis of an LP-based method. Oper Res 54:419–435
Buist E, L'Ecuyer P (2005) A java library for simulating contact centers. In: Proceedings of the winter simulation conference, 2005, p 10
Çezik M, L'Ecuyer P (2008) Staffing multiskill call centers via linear programming and simulation. Manag Sci 54:310–323
Chan W, Koole G, L'Ecuyer P (2014) Dynamic call center routing policies using call waiting and agent idle times. Manuf Serv Oper Manag 16:544–560
Chan W, Ta TA, L'Ecuyer P, Bastin F (2016) Two-stage chance-constrained staffing with agent recourse for multi-skill call centers. In: The 2016 winter simulation conference, IEEE Press, Piscataway, NJ, USA, pp 3189–3200
Excoffier M, Gicquel C, Jouini O (2016) A joint chance-constrained programming approach for call center workforce scheduling under uncertain call arrival forecasts. Comput Ind Eng 96:16–30 (**ISSN 0360-8352**)
Excoffier M, Gicquel C, Jouini O, Lisser A (2015) Comparison of stochastic programming approaches for staffing and scheduling call centers with uncertain demand forecasts. In: Pinson E, Valente F, Vitoriano B (eds) Operations research and enterprise systems. Springer, Cham
Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: tutorial, review, and research prospects. Manuf Serv Oper Manag 5:79–141
Gurvich I, Luedtke J, Tezcan T (2010) Staffing call centers with uncertain demand forecasts: a chance-constrained optimization approach. Manag Sci 56:1093–1115
Hoeffding W (1994) Probability inequalities for sums of bounded random variables. The collected works of Wassily Hoeffding. Springer, pp 409–426
Ingolfsson A, Campello F, Wu X, Cabral E (2010) Combining integer programming and the randomization method to schedule employees. Eur J Oper Res 202:153–163
Jouini O, Koole G, Roubos A (2012) Performance indicators for call centers with impatience. IIE Trans 45:01
Kooley G, Nielsen BF, Nielsen T (2015) Optimization of overflow policies in call centers. Probab Eng Inf Sci 29:07
L'Ecuyer P, Meliani L, Vaucher J (2003) SSJ: a framework for stochastic simulation in java. vol 1, pp 234–242 vol 1, 01. ISBN 0-7803-7614-5
Nemirovski A, Shapiro A (2006a) Convex approximations of chance constrained programs. SIAM J Optim 17:969–996. https://doi.org/10.1137/050622328
Nemirovski A, Shapiro A (2006b) Scenario approximations of chance constraints. Probabilistic and randomized methods for design under uncertainty. Springer, pp 3–47
Singh B, Watson J-P (2019) Approximating two-stage chance-constrained programs with classical probability bounds. Optim Lett 13:9
Ta TA, Chan W, Bastin F, L'Ecuyer P (2019) A simulation-based decomposition approach for two-stage staffing optimization in call centers under arrival rate uncertainty. Working paper. https://www.iro.umontreal.ca/~lecuyer/myftp/papers/ssa-2stage-decomp.pdf
Ta TA, Mai T, Bastin F, L'Ecuyer P (2020) On a multistage discrete stochastic optimization problem with stochastic constraints and nested sampling. Math Progr 190:1–37
Wallace R, Whitt W (2005) A staffing algorithm for call centers with skill-based routing. Manuf Serv Oper Manag 7:276–294
Xie W, Ahmed S, Jiang R (2017) Optimized bonferroni approximations of distributionally robust joint chance constraints. Optimization Online, 02
Zan J, Hasenbein J, Morton D (2014) Asymptotically optimal staffing of service systems with joint QoS constraints. Queueing Syst 78:04. https://doi.org/10.1007/s11134-014-9406-x