Abstracting and optimally staffing a real-life call center with multi-skill priority routing

Julián Ramón Marrades Furquet
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—Thanks to our collaboration with a private company, we show how to abstract a multi-skill priority routing call center by means of call data, and how to assess the quality of the resulting model. Then, an existing staffing algorithm based on Lagrangian duality is adapted for the firm's use case. Our method is tested against the latest publication, which employs a population-based evolutionary algorithm inspired by elephant herds. We find that that, while the elephant-based approach is 4 times faster than our method, we are able to produce minimal cost staffings with lower variance and similar quality to our quick competitor. Moreover, possible improvements to both procedures are pointed out.**
To conclude, we compare the output of the algorithms to the staffing which the firm is currently employing.

***Index Terms*—stochastic optimization, nonlinear programming, skill-based routing, priority routing, local search, Lagrangian duality, Elephant Herding Optimization**

I. Introduction

The first call center originates around 1965, when The Birmingham Press and Mail created a small team which handled customer contacts via telephone [1]. Since then, a lot of things have changed: some centers employ thousands of agents, and tools such as artificial intelligence are often used for a more effective customer service [2]. Such developments, and many others, are the reason why the market size of call centers has significantly increased [3] in recent years, and together with it the so-called contact center software market, which is expected to keep growing in the near future [4]. One can safely predict that call centers will become more relevant for the consumer society as the time goes by.

In the world of academia, call centers have always been a popular topic in the sector of operations research. The literature can be divided into three separate blocks:

1) modeling and/or performance analysis of the underlying queuing system (see, e.g., [5]–[10]),
2) analytically approximating performance measures instead of estimating them by means of simulation (see, e.g., [6], [11]–[13]), and
3) optimization, which can be subdivided into two different areas:
   a) routing optimization, which targets the Automatic Call Distributor (ACD) [14], [15], i.e., deciding (i) what to do with an incoming call and (ii) what to do with an agent who just became idle (see, e.g., [16], [17]); and
   b) staffing optimization, which tries to minimize the cost of the set of agents serving the calls while still providing an acceptable customer service. Staffing problems can get even more complex if we impose shift restrictions (allowed working times) and/or have different periods in which the definition of *acceptable customer service* changes, and the call arrival and handling rates are different (see, e.g., [18]–[22]).

This paper emerges from the collaboration with a firm which runs a medium-sized call center. In the system, there are different types of calls and different groups of agents. Each group is defined by a set of call types (skill set) which every agent from that group can handle.
The company was interested in minimizing the cost of their staffing while ensuring a satisfactory performance for each call type. For that purpose, we were provided with all call data from 2019, and asked to create a queuing model and solve a single-period staffing problem with unrestricted shifts. In other words, we seek the number of agents per group as if they can work non-stop and the call arrival and service rates do not change.

Existing research has mainly targeted single-skill call centers, in which there is a single type of call (see, e.g., [23], [24]). Nowadays, companies, including our partner, have started to cross-train their agents so that they can handle different types of customers. For instance, different languages or different types of products. Then, we have *specialists*, i.e. agents which can only handle one type of customer, and *generalists*, i.e. multi-skilled agents. This cross-training has brought a new level of difficulty into staffing problems. Consequently, several algorithms for multi-skilled staffing have been published in recent years, which we will discuss and build upon in this paper.

Overall, we make 2 key contributions:

1) While staffing literature frequently assumes that call arrival and service rates are given, we derive them

This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Knowledge Engineering, Maastricht University. Supervisor(s): Matúš Mihalák, Gijs Schoenmakers, and Joël Karel.

from the provided data and assess the accuracy of our inferences.

2) We adapt the approach of Pot et al. (2008) [25] to solve staffing problems with multiple performance constraints. The adaptation clears up the doubt of Avramidis et al. (2009) [26] whether the method of Pot. et al. (2008) [25] can be adapted to work in such type of scenarios. Moreover, we compare our algorithm to the work of Horng and Lin (2020) [27], which is the most recent published paper on multi-skilled staffing.

Hereinafter, this paper is organized as follows. Section II provides guidelines to model a multi-skilled call center given a generic input data format. Section III gives a formal definition of the multi-skilled staffing problem. Section IV examines the existing literature for (sub)optimally staffing such type of call centers. Section V explains our adaptation of the method proposed by Pot et al. (2008) [25]. Section VI presents some experiments to analyze the performance of our algorithm, and to compare it with the most recent procedure. Section VII discusses the results of the experiments, paying special attention to areas where improvements can be made. Finally, VIII concludes this paper.

II. Abstracting the call center

We received tabular data in which each row is a call instance with the following attributes: call type, arrival timestamp, handled (boolean), agent type, delay, and handling time, with the last two having seconds as unit.

In this section, we guide the reader through the steps we followed to convert the original dataset into a queuing model. We do not mention the ACD because the routing policy is hardcoded as given by the firm.

A. Data inspection and cleaning

We start by performing a sanity check. Do the attribute values make sense? E.g. are there negative delays or handling times? If so, one needs to make adjustments before proceeding into further stages.

Then, we look for rows which miss essential attributes and discard them. For instance, if a row is missing the agent group, arrival time, etc., we cannot use it to create a queuing model.

Afterwards, with input from the domain experts, we check if any agent has handled a call which is not within her skill set. This is the case for only 4 entries in the dataset, so we remove them.

Finally, after conversations with the firm, we discard every call which has a lower handling time than 30 seconds, since most of those represent connection cuts or other minor incidences.

B. Fitting probability distributions

Fig. 1 shows a diagram for a generic multi-skill multi-queue call center. We observe three areas in the diagram that we must model:

1) Interarrival times: how fast are the calls arriving.
2) Handling times: how fast are the calls being served.

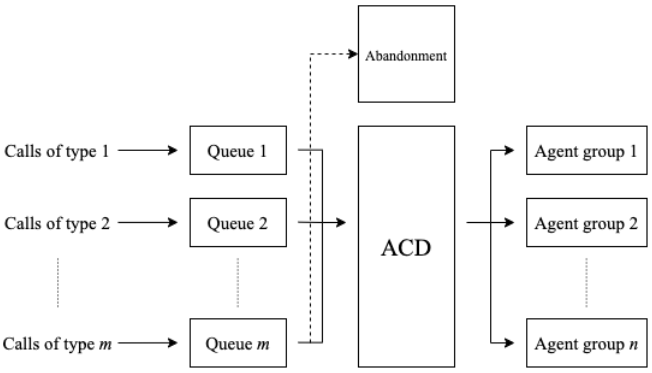


Fig. 1: Flow diagram for a multi-skilled multi-queue call center. Note that each agent group is a set of agents itself, and all agents within a group can handle the same set of call types.

3) Abandonment times: how long are customers willing to wait in the queue before they hang up the phone.

The most common way to model these areas is by fitting probability distributions to the raw data and sampling from them during the simulation. Scientific computing packages such as Scipy, which we use in this paper, provide the tools to fit many probability distributions to raw observations by means of Maximum Likelihood Estimation (MLE) [28], [29]. While most of the existing literature assumes arrival, handling and abandonment to be stationary Poisson processes, we will look at the shape of the data and choose a probability distribution accordingly, for a more realistic model.

Firstly, we check if we have the data needed to model abandonment. We look at those calls which were not handled and realize that their delay is missing. After a conversation with the domain experts, we realize that we are not able to model customer loss since the data does not show how long they had been waiting for before they left.

Then, we turn our attention to call arrivals. After plotting the interarrival times for each call type, we decide that fitting an exponential distribution is adequate (see Fig. 3a for an example)

Lastly, we generate histograms for the service times of the different (call type, agent group) tuples and reach the conclusion that a gamma distribution may accurately fit the data (see Fig. 3b for an example).

After fitting the distributions, we check if there is relationship between the number of data-points in the dataset and the goodness-of-fit of the MLE estimation. For that purpose, we employ the Kolmogorov-Smirnov (K-S) test [30], which measures the maximum distance between the empirical and fitted cumulative distribution function (CDF). Such distance is called the K-S statistic.

On the one hand, in Fig. 2a we observe an improvement in the quality of the fit with the increase of the number of entries in the dataset, which is something to expect if the observations really belong to the hypothesized distribution.

On the other hand, Fig. 2b depicts a slightly different scenario.

For low quantities of data, the goodness-of-fit shows a lot of variance, due to the fact that accurate parameter estimation usually requires thousands of datapoints. For larger amounts of calls, the K-S statistic seems to stabilize in the range $[0.57, 0.67]$ instead of decreasing. This phenomena may be due to

1) the higher number of parameters to estimate for the gamma distribution,
2) very high service times (outliers) to which the statistic is sensitive to, and
3) gamma not really being the underlying distribution.

All in all, when the system is simulated with a staffing provided by the domain experts and the results are tested against the real observations, both the arrival and service times satisfy the requirements of the firm. Fig. 3 shows that the sampled observations are close to the real ones, even when the K-S statistic is large. Hence, we can conclude that testing the system against unseen data may be a better way to assess the quality of the model than goodness-of-fit statistical tests.

III. Problem formulation

Let there be n groups of agents and m types of calls. Each call type $j \in \{1, 2, \ldots, m\}$ has its own queue, an exponentially-distributed interarrival time given by the probability density function (PDF) $h_j(t)$, and a priority score denoted by p_j. The weight matrix W expresses the priority that each agent group has for each type of call, i.e. W_{ij} is the priority score of an agent from group i for call type j. It can be the case that a certain group i^* cannot handle a given type of call j^*; then, $W_{i^*j^*} = M$, where M is an arbitrarily large number. For the (i, j) pairs where $W_{ij} \neq M$, the service time of an agent from group i when handling call type j follows a gamma distribution given by the PDF $g_{ij}(t)$, i.e. the handling times are group- and type-dependent.

By solving the staffing problem, we aim to determine how many agents per group should be allocated so that:

1) the total cost of the staffing is minimized, in our case defined by the number of agents in the staffing, and
2) the following service level (SL) constraint is met: the average delay of each call type must be no higher than a given threshold. This includes all calls, not only steady-state.

If we let $\mathcal{D}_j$ be the set of delays of call type j, then we can express the SL constraint as follows:

$$\mathrm{SL}_j = \frac{\sum_{d \in \mathcal{D}_j} d}{|\mathcal{D}_j|} \leq \tau_j, \tag{1}$$

where τ_j is the target average for call type j.

The firm's ACD can be described in terms of what to do when (1) a new call arrives and (2) an agent becomes idle.
(1) When a new call arrives, say type j', it is directed to its respective queue. If the queue is not empty, it means that there are earlier calls which should be handled first, and thus the new call will remain there. However, if there are no previous calls waiting, the idle agents that can handle j'-calls are sorted in ascending order by their priority score for call type j'. Then, out of those with minimum score, the call is given to the one who has been idle the longest. Formally, let b_k be the amount of time that agent k has been idle, where $k \in \{1, 2, \ldots, a\}$ and a denotes the total number of agents in the system. We will use $b_k = -1$ to indicate that the agent is currently busy. Moreover, let $\mathcal{G}(k)$ be a function that returns the group of k. Then, the agent k^*, which will receive the call, can be written as:

$$k^* = \arg\max_{k'} b_{k'}, \quad k' \in \underset{k \mid b_k \neq -1 \wedge W_{\mathcal{G}(k)j'} \neq M}{\arg\min} W_{\mathcal{G}(k)j'} \tag{2}$$

(2) When an agent k' becomes idle, the calls in the system whose type can be handled by the agent are sorted in ascending order of priority score. Then, out of those with minimum score, the one which has been in queue the longest is selected. Formally, let q_l be the amount of time that call l has been in queue, where $l \in \{1, 2, \ldots, c\}$ and c represents the total amount of calls in the system at that particular moment. Also, let $\mathcal{T}(l)$ be a function that returns the type of l. Then, the selected call l^* can be written as:

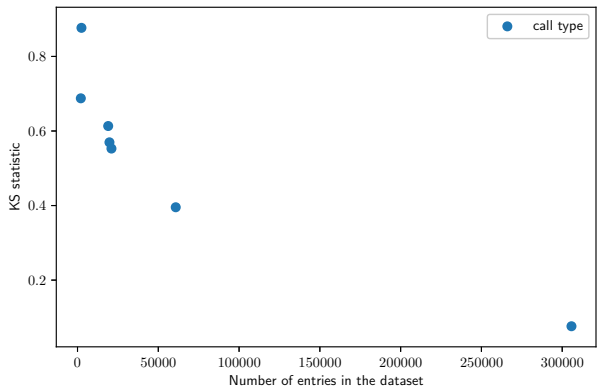
$$l^* = \arg\max_{l'} q_{l'}, \quad l' \in \underset{l \mid W_{\mathcal{G}(k')\mathcal{T}(l)} \neq M}{\arg\min} p_{\mathcal{T}(l)} \tag{3}$$

IV. Prior work

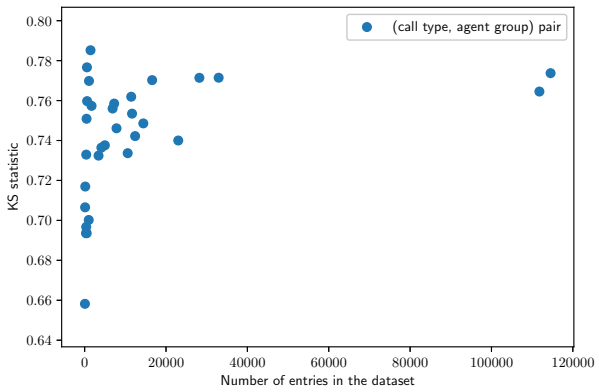
In this section, we will explore the most relevant multi-skill staffing methods and some of the most recent publications. It is relevant to state that (1) all methods are heuristic, and (2) key assumptions are made with respect to the convexity of the problem. Nonetheless, the algorithms are designed to ensure that such assumptions are very likely to hold.

Harrison and Zeevi (2005) [31] develop a staffing optimization algorithm for non-stationary arrival rates based on a combination of linear programming and Monte Carlo simulation. One of their contributions is the presence of an average approximation of a SL constraint in their objective function. In particular, they try to close the gap between the arrival rate for call type j and the total service rate of the staffing vector $\vec{x}$ for that call type. Koole and Pot (2006) [15] take that average performance estimator and include it as a constraint in an integer linear program. Such program tries to minimize staffing costs while making sure that the total service rate for call type j is not smaller than its arrival rate. This results in a staffing vector which may not satisfy all call type SL constraints, but is a good starting point for a local search algorithm.

Wallace and Whitt (2005) [32] assume that service rates only type-dependent, and allow the agents to possess any set of skills. Even though such assumptions are not very realistic, they make several key contributions.
Firstly, they study the effect that the number of skills per agent has in the performance of the system. They conclude that the performance of a staffing of 2-skill agents is comparable to the performance when agents possess all skills. To be able to compare their staffing result with a theoretical optimum,

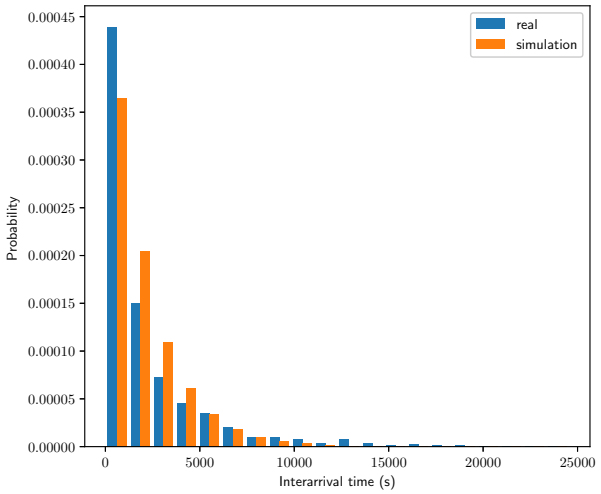


(a) Exponential fit in arrival times. Each data-point represents a different call type.

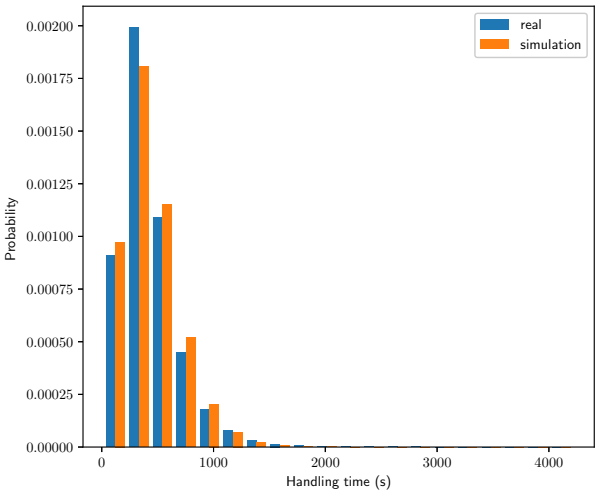


(b) Gamma fit in service times. Each data-point represents a (call type, agent group) pair.

Fig. 2: Assessment of the probability distribution fits for arrival and service times by means of the Kolmogorov-Smirnov (K-S) test.



(a) Interarrival times for call type 6. The model was trained on 2438 datapoints with a resulting statistic K-S $= 0.877$.



(b) Handling times for agent group 12 and call type 6. The model was trained on 1693 datapoints with a resulting statistic K-S $= 0.757$.

Fig. 3: Comparison among real observations and simulated samples of interarrival and handling times.

they determine the minimum amount of *superagents* (agents with all skills) that is required to satisfy the SL constraints. Afterwards, an extension of the square-root-safety-staffing rule is used to create an initial staffing. Finally, they employ a simple local search algorithm to ensure that the constraints are met.

Cezik and L'Ecuyer (2008) [33] build upon the cutting-plane method developed by Kelley Jr. (1960) [34], and adapted for staffing of single-call-type single-skill call centers by Atlason et al. (2004) [35]. They formulate a stochastic inequality constrained optimization program (SICOP), i.e. where at least one of the constraints is stochastic, and estimate such constraints via lengthy simulations. When the algorithm generates a staffing $\vec{x}'$ which does not meet a SL constraint for call type j, a linear constraint is generated so that $\vec{x}'$ is excluded from the set of feasible solutions without removing any other feasible solution for the original problem if $\mathrm{SL}_j(\vec{x})$ is convex. The main issue of this method is the large computational expense needed to generate the linear cuts.

Chevalier and Van den Schrieck (2008) [36] reduce the queuing model to a blocking system, i.e. where if a customer cannot be served immediately after arrival, he leaves the system. Thanks to the simplification, performance approximation techniques developed by Fredericks (1980) [37] and Tabordon (2002) [38] can be used to avoid simulation. Then, a branch-and-bound-type search algorithm is proposed, which consists of partial enumeration of the solution space, limiting the size of the problems which can be handled. The key

drawback is the assumption that service times are exponential and independent of call type and agent group.

Pot et al. (2008) [25] propose a more elegant SICOP than Cezik and L'Ecuyer (2008) [33], in which they aim to fight the curse of dimensionality by reducing the search space. This way, their algorithm is able to solve larger problems. The Lagrangian dual of the proposed SICOP is formulated and then solved by a heuristic method that exploits concavity to perform an efficient search. It was observed, by comparing simulation to approximated blocking models, that the latter tend to yield a larger amount generalists. This is expected because such models want to ensure that a customer can be served upon arrival, resulting in agents with larger skill sets. A limitation of this method is that it only supports a global service level constraint, independent of call type.

Avramidis et al. (2009) [26] expand the work of Koole and Talim (2000) [39] to develop an approximation of the SL per call type. It expands the traditional blocking model by incorporating queuing, and assuming that if a call of type j is put in queue, it will wait until an agent which has j as his lowest priority call type can handle it, i.e. a worst case scenario analysis. They also provide a heuristic approach to create an initial staffing and a simulation-based local search algorithm to ensure feasibility. Such local search estimates the service level per type and the rate of j-calls being completed by agent group i. Then, it uses that information to (1) add an agent to the group that has the most impact on type $j*$ if that SL constraint is violated, and (2) remove an agent if all constraints are satisfied.

Chan et al. (2016) [40] extend the linear cut method of Cezik and L'Ecuyer (2008) [33] to solve a staffing problem where the arrival rates are uncertain. Their work is improved by Ta et al. (2019) [41], who starting from the work of Harrison and Zeevi (2005) [31], formulate a two-stage SICOP and employ the Monte Carlo method to simulate scenarios under different realizations of the arrival rates.

Horng and Lin (2020) [27] employ Elephant Herding Optimization (EHO) with very short simulations to generate a tentative set of candidate staffings and then select the best solution from it. EHO is a population-based optimization technique, proposed by Wang et al. (2016) [42], which mimics herding behavior. It has also been successful in other areas such as emotion recognition problems [43].

V. Algorithm

Let $\vec{x} = [x_1, \ldots, x_n]^T$ and $\vec{c} = [c_1, \ldots, c_n]^T$ be the staffing and cost vectors, respectively. Then, let $\vec{\tau} = [\tau_1, \ldots, \tau_m]^T$ denote the target average delay for each call type. We write $\vec{\text{SL}}(\vec{x}) = [\text{SL}_1(\vec{x}), \ldots, \text{SL}_m(\vec{x})]^T$, where $\text{SL}_j(\vec{x})$ stands for the average delay of call type j under staffing $\vec{x}$ (see Eq. 1).

Almost identically to Pot et al. (2008) [25], our SICOP can be written as follows:

$$\begin{aligned} \min_{C \in \mathbb{Z}_+} & \left(\min_{\vec{x} \in \mathbb{Z}_+^m : \vec{x} \cdot \vec{e} = C} \vec{c} \cdot \vec{x} \right) \\ \text{s.t.} \quad & \vec{\text{SL}}(\vec{x}) \leq \vec{\tau}, \end{aligned} \tag{4}$$

where $\mathbb{Z}_+$ represents the set of non-negative integers, C denotes the total number of agents, and $\vec{e}$ is a vector with a 1 at every position. In other words, this two-stage optimization aims to find the minimum cost staffing for all C, to later select the best out of those.

We formulate the Lagrangian function as

$$f\left(\vec{\beta}, \vec{x}\right) = \vec{\beta} \cdot \left(\vec{\text{SL}}(\vec{x}) - \vec{\tau} \right) + \vec{c} \cdot \vec{x}, \tag{5}$$

where $\vec{\beta}$ denotes the Lagrange multiplier vector.
Then, for each C, we write the Lagrange dual function as

$$f^{(C)}\left(\vec{\beta}\right) = \min_{\vec{x} \in \mathbb{Z}_+^m : \vec{x} \cdot \vec{e} = C} f\left(\vec{\beta}, \vec{x}\right) \tag{6}$$

Hence, our initial two-stage program can be stated as

$$\begin{aligned} & \min_C f^{(C)}, \quad \text{where} \\ & f^{(C)} := \max_{\vec{\beta} \in \mathbb{R}_+^m} f^{(C)}\left(\vec{\beta}\right) \end{aligned} \tag{7}$$

We also define $\vec{\text{SL}}$ for C agents under a given $\vec{\beta}$ by

$$\vec{\text{SL}}^{(C)}\left(\vec{\beta}\right) := \vec{\text{SL}}\left(\operatorname*{arg\,min}_{\vec{x} \in \mathbb{Z}_+^m : \vec{x} \cdot \vec{e} = C} f\left(\vec{\beta}, \vec{x}\right) \right) \tag{8}$$

Namely, $\vec{\text{SL}}^{(C)}\left(\vec{\beta}\right) = \vec{\text{SL}}\left(\vec{x}^*\right)$, where $\vec{x}^*$ has a total of C agents and minimizes Eq. 5 under the given $\vec{\beta}$.

Then, we adjust the heuristic proposed by Pot et al. (2008) [25] as indicated in Algorithm 1.

In lines 8-13, we optimize Eq. 6 by iterating over staffings of C agents following the local search method of Pot et al. (2008) [25]. Specifically, we employ simulation to estimate $\vec{\text{SL}}(\vec{x})$, contrary to the performance approximation approach used in the original paper. In lines 14-22 we maximize $f^{(C)}$ by performing bisection over $\vec{\beta}$. In lines 24-27, we solve Eq. 7 by iterating over all C by golden ratio search.

In order to safely employ bisection and golden ratio search, the functions which we are optimizing are required to be, at least, monotonic.
On the one hand, since Pot et al. (2008) [25] only formulate a single constraint $\text{SL}(\vec{x}) \leq \tau$, i.e. there is just one Lagrange multiplier β, it is safe to assume that $f^{(C)}(\beta)$ monotonically decreases as β increases, for a fixed C. The reason is that, with an increase in β, the algorithm is forced to find a staffing that further minimizes $\text{SL}(\vec{x}) - \tau$, which at the same time would minimize $f(\beta, \vec{x})$. Moreover, it is assumed that $f^{(C)}$ is convex for $C \in [C_L, C_U]$. This is very likely since staffing costs are typically linear and $\text{SL}^{(C)}(\beta)$ usually decreases as C increases, i.e. it is monotonic w.r.t. C.
On the other hand, for several SL constraints, such assumptions are not so trivial. While it is true that for a given j, $\vec{\text{SL}}_j^{(C)}\left(\vec{\beta}\right)$ is monotonic w.r.t. $\vec{\beta}_j$ by the same reasoning above, the effect of $\vec{\beta}_j$ on other SLs, and therefore the impact on $f^{(C)}\left(\vec{\beta}\right)$, is difficult to predict. Thus, we propose to only adjust one $\vec{\beta}_j$ at a time, giving the opportunity to fight the side-effects in the next iteration (see Algorithm 1). Then, we

Algorithm 1 Staffing heuristic by Pot et al. (2008), adjusted

Input: ϵ: threshold for Lagrange multiplier convergence
C_L, C_U: lower and upper bound for the number of agents, respectively
Output: $\vec{x}^{***}$: the staffing vector

Initialization: $\vec{x}^{***} \leftarrow \vec{0}$, $f \leftarrow M$, $f^{(C)} \leftarrow 0$, $\vec{\beta}_L \leftarrow \vec{0}$, $\vec{\beta}_U \leftarrow \vec{M}$
for $C \in \{C_1, C_2, C_3, C_4\}$, determined by golden ratio search on $[C_L, C_U]$ **do**
  Init $f^{(C)}\left(\vec{\beta}\right) \leftarrow 0$, $x^{**} \leftarrow \vec{0}$ and $\vec{\beta} \leftarrow \frac{1}{2}\left(\vec{\beta}_L + \vec{\beta}_U\right)$
  while $\exists j \in \{1, \ldots, m\} : \vec{\beta}_{U_j} - \vec{\beta}_{L_j} > \epsilon$ **do**
    $j^* \leftarrow \arg\max_j \left(\vec{\beta}_{U_j} - \vec{\beta}_{L_j}\right)$
    Init $f^{(C)}\left(\vec{\beta}\right) \leftarrow M$ and $\vec{x}^* \leftarrow \vec{0}$
    Init $\vec{\mathrm{SL}}^{(C)}\left(\vec{\beta}\right) \leftarrow 0$
    for all $\vec{x} : \vec{x} \cdot \vec{e} = C$ **do**
      Calculate $\vec{\mathrm{SL}}(\vec{x})$ and $f\left(\vec{\beta}, \vec{x}\right)$
      if $f\left(\vec{\beta}, \vec{x}\right) < f^{(C)}\left(\vec{\beta}\right)$ **then**
        $\vec{x}^* \leftarrow \vec{x}$, $f^{(C)}\left(\vec{\beta}\right) \leftarrow f\left(\vec{\beta}, \vec{x}\right)$ and $\vec{\mathrm{SL}}^{(C)}\left(\vec{\beta}\right) \leftarrow \vec{\mathrm{SL}}(\vec{x})$
      end if
    end for
    if $f^{(C)}\left(\vec{\beta}\right) > f^{(C)}$ **then**
      $\vec{x}^{**} \leftarrow \vec{x}^*$ and $f^{(C)} \leftarrow f^{(C)}\left(\vec{\beta}\right)$
    end if
    if $\vec{\mathrm{SL}}_{j^*}^{(C)}\left(\vec{\beta}\right) > \vec{\tau}_{j^*}$ **then**
      $\vec{\beta}_{L_{j^*}} \leftarrow \vec{\beta}_{j^*}$
    else
      $\vec{\beta}_{U_{j^*}} \leftarrow \vec{\beta}_{j^*}$
    end if
    $\vec{\beta} \leftarrow \frac{1}{2}\left(\vec{\beta}_L + \vec{\beta}_U\right)$
  end while
  if $f^{(C)} < f$ **then**
    $f \leftarrow f^{(C)}$ and $\vec{x}^{***} \leftarrow \vec{x}^{**}$
  end if
  Update C_L, C_U by golden ratio.
end for
return $\vec{x}^{***}$

are left with the convexity of $f^{(C)}$. Even though the linearity of the costs can still be assumed, the monotonicity of $\vec{\mathrm{SL}}^{(C)}\left(\vec{\beta}\right)$ w.r.t. C is complicated to assess. This is because a staffing of C' agents may perform worse than a staffing of $C'' = C' - 1$ agents if the group sizes are not adequate. However, since the algorithm will try to find a proper staffing for a given C, we can state that $\vec{\beta} \cdot \left(\vec{\mathrm{SL}}^{(C)}\left(\vec{\beta}\right) - \vec{\tau}\right)$ should decrease as C increases.

Despite all uncertainties regarding convexity, the algorithm performs well in practice, as we will expose in Section VII.

To determine C_L and C_U we create a generalist and a specialist composition.
The generalist composition contains a single agent group which has the capability to handle all types of calls at maximum speed, i.e. its handling time distribution $g_{Gj^*}(t)$ for a given call type j^* is given by

$$g_{Gj^*}(t) = g_{i^*j^*}(t), \quad i^* = \underset{i \mid W_{ij^* \neq M}}{\arg\min} \mathbf{E}\left[g_{ij^*}(t)\right], \tag{9}$$

where $\mathbf{E}[\cdot]$ denotes expectation. Then, we progressively increase the number of agents in the generalist group until all service levels are satisfied, finding our lower bound.
The specialist composition is made up of m agent groups, where agents of group j^{**} can only handle calls of type j^{**} and at minimum speed, i.e. their handling time distribution $g_{Sj^{**}}(t)$ is

$$g_{Sj^{**}}(t) = g_{i^{**}j^{**}}, \quad i^{**} = \underset{i \mid W_{ij^{**} \neq M}}{\arg\max} \mathbf{E}\left[g_{ij^{**}}(t)\right] \tag{10}$$

Starting with 1 agent per specialist group, we iteratively increase the size of group j^{**} by 1 when its SL constraint is not met until all requirements are satisfied, obtaining the upper bound.

VI. Experiments

Our experiments are performed for the firm's use case, where we find $n = 12$ and $m = 7$. Let $\vec{p} = [3, 2, 2, 1, 1, 2, 1]^T$ denote the priority vector for call types. The agreed delays per call type are $\vec{\tau} = [160, 45, 45, 45, 45, 45, 45]^T$. The cost vector is $\vec{c} = \vec{e}$, i.e. the cost of the staffing is the total number of agents. The weight matrix W is

$$W = \begin{bmatrix} 10 & M & M & M & M & M & M \\ 10 & M & M & M & 5 & M & M \\ 20 & 15 & 15 & 5 & M & 15 & M \\ 20 & 15 & 15 & M & 5 & 15 & M \\ M & 10 & 10 & 5 & M & 10 & M \\ M & M & M & M & M & 1 & M \\ M & 10 & 10 & M & M & 10 & M \\ M & M & M & 1 & M & M & M \\ M & M & M & M & 1 & M & M \\ M & M & 1 & M & M & M & M \\ 20 & 15 & 15 & M & M & 15 & 5 \\ 10 & M & M & M & M & M & 5 \end{bmatrix}$$

When running the EHO method, the parameters suggested in the original paper [27] are employed. For the Lagrangian duality (LD) approach, we use $\epsilon = 0.1$ and set all upper bounds for Lagrange multipliers to 50. For our use case, we find $C_L = 24$ and $C_U = 42$.

In Section VII-A, we explore whether $f^{(C)}$ and $f^{(C)}\left(\vec{\beta}\right)$ are convex/monotonic by running the LD algorithm several times and gathering metadata. For $f^{(C)}$, we can simply plot its value for different C. However, we cannot graph $f^{(C)}\left(\vec{\beta}\right)$ for different $\vec{\beta}$ because $\vec{\beta}$ is 7-dimensional. Hence, we will plot $f^{(C)}\left(\vec{\beta}\right)$ as the algorithm performs adjustments on $\vec{\beta}$ for a

given C. If we observe that $f^{(C)}\left(\vec{\beta}\right)$ monotonically increases over time, we can conclude that it should be monotonic w.r.t. $\vec{\beta}$.

We compare our adaptation of the LD approach to the EHO procedure of Horng and Lin (2020) [27] by performing 2 experiments:

1) We run each algorithm 5 times, saving the staffing returned by the algorithm and the hourly best (see Section VII-B). Out of the 5 runs, we select the minimum cost solution provided by each method for further evaluation in the next experiment.
2) We employ both minimum cost solution runs to assess the quality of the algorithms if one were to have execution time restrictions (see Section VII-C).

We will also compare the solutions of the algorithms of the current staffing of the firm in Section VII-D.

VII. Discussion

A. Experiment 1: Convexity and monotonicity

Fig. 4 shows the shape of $f^{(C)}$ in the range $[C_L, C_U]$. In Fig. 4a we observe several outliers for $C = 24$ and $C = 25$, which indicates that the algorithm struggles to deal with small C. The reason may be that it is really not possible to find a feasible staffing for such values of C, or that $f^{(C)}$ is not convex in that range. Nonetheless, if we discard the outliers, we observe a defined convex shape, as depicted in Fig. 4b.

In Fig. 5, we can see the evolution of $f^{(C)}\left(\vec{\beta}\right)$ as the algorithm adjusts $\vec{\beta}$ for a given C. We observe a continuous increase for $C = 27$, $C = 28$, $C = 35$, and $C = 42$, indicating that $f^{(C)}\left(\vec{\beta}\right)$ is monotonic w.r.t. $\vec{\beta}$ for those number of agents. However, for $C = 24$ the plot shows that, even though there is a slight increase in $f^{(C)}\left(\vec{\beta}\right)$, it is not continuous. This phenomena, together with the outliers present in Fig. 4a, further indicate that the proposed algorithm does not work so well for small values of C, i.e. when C is very close to C_L. To fight this problem, one could slow down the convergence of $\vec{\beta}$, since its adjustments seem to have a more unpredictable impact on $f^{(C)}\left(\vec{\beta}\right)$ for $C = 24$ than for larger values of C.

B. Experiment 2: EHO vs. LD

By looking at Table I, one may notice that, while EHO provides the best solution, the variance in its costs, ranging from 26 to 35, is higher than the costs provided by LD, which are in the range $[27, 30]$. This phenomena is due to the randomness present in population-based optimization methods. Such randomness can also generate a final optimal staffing that is worse than the best agent composition of a previous iteration. Even though Horng and Lin (2020) [27] do not mention this effect, it would be ideal to progressively save the best staffing every I iterations to increase the probability of obtaining better solutions.

If one looks at the staffing provided by both algorithms for a fixed cost, e.g. 28, one will observe that, for most call types, the LD solution provides far better SLs than the EHO result. The reason is as follows: while EHO has no knowledge of the problem, i.e. it just evaluates staffings and ranks them, LD adjusts the agent compositions so that call types with higher average delay are prioritized.

Another relevant difference is that LD tends to use less agent groups for its optimal staffing, and those groups include agents with a large skill sets. Since we are employing a cost function which essentially targets the number of agents, LD tends to create compositions with few agents that can handle many types of calls. However, in reality, the cost of an agent is usually determined by the number of skills she has. If we were to employ such cost function, we would expect a much more homogeneous mixture of specialists and generalists.

C. Experiment 3: EHO vs. LD over time

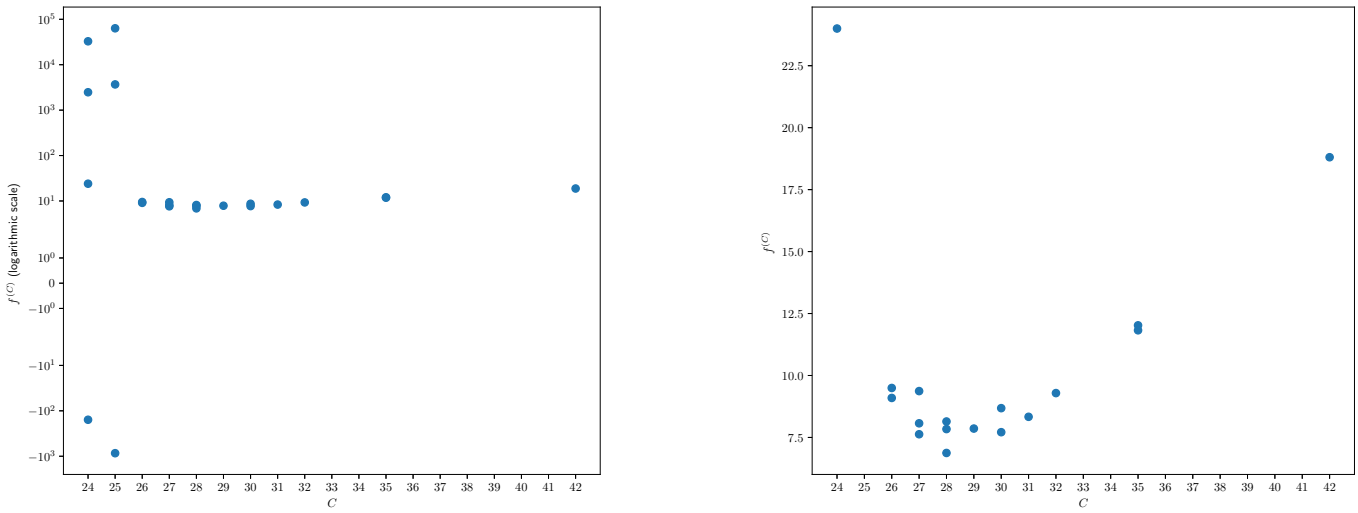
The results are presented in Fig. 6 and Table II. On the one hand, the LD approach takes 2 hours to come up with an initial staffing, and 4 hours to generate a feasible solution. Afterwards, it keeps improving until it converges to a final optimum after 17 hours of execution. On the other hand, the EHO procedure generates an almost feasible solution after 1 hour of run time, and reaches a very low cost feasible staffing even before LD provides one which satisfies all SL constraints. What is more, it terminates after 4 hours, which makes EHO much faster than our algorithm. Hence, EHO is a better method when one does not have 17 hours to spare.

To speed up our LD adaptation, we identify three points of improvement:

1) Employ state-of-the art performance estimators to calculate the SLs for a given agent composition, trading a lower execution time for a probable higher-cost staffing.
2) Study the effect that changing component j of the Lagrange multiplier vector has in $\vec{\text{SL}}_{j'}$, $j' \neq j$. One could exploit such side-effects to adjust more than one multiplier at a time, speeding up the convergence significantly.
3) Better implementation. If the cost of the staffing is the number of agents, not exploring staffings for C agents when a feasible agent composition has been found for $C' < C$ would reduce the execution time.

D. Experiment 4: Firm vs. Algorithms

The current staffing of the firm is displayed in Table III. According to our simulation, most calls are answered as soon as they arrive in the system. However, this is not what we observe in reality, because agents are not working non-stop: they take coffee breaks, have lunch, get sick, etc. This is the reason why there is a huge gap between the cost of the current staffing and the costs of the solutions provided by EHO and LD. Unfortunately, we were not given data to model such interruptions. Therefore, how to translate the staffings from the algorithms into *real-life* agent compositions remains a question to be answered. One could try to find a relationship between *virtual* and *real* staffings, or develop a more complex model which takes into account all possible side events. The first option should be a better choice since a sophisticated model



(a) Raw data

(b) Without outliers

Fig. 4: Scatter of $f^{(C)}$ for different C, over several LD algorithm runs.

TABLE I: Comparison between the Elephant Herding Optimization method proposed by Horng and Lin (2020) [27] and our adaptation for the Lagrangian duality approach of Pot et al. (2008) [25]. Minimum cost staffing satisfying all SL constraints.

Method	**Cost**	**Staffing**	$\text{SL}_1 - \tau_1$	$\text{SL}_2 - \tau_2$	$\text{SL}_3 - \tau_3$	$\text{SL}_4 - \tau_4$	$\text{SL}_5 - \tau_5$	$\text{SL}_6 - \tau_6$	$\text{SL}_7 - \tau_7$
EHO	26	$[2, 5, 5, 5, 2, 1, 1, 1, 2, 0, 3, 4]^T$	-23.450	-15.180	-15.933	-13.795	-29.171	-15.320	-20.839
	28	$[2, 5, 3, 4, 2, 1, 1, 2, 0, 0, 5, 3]^T$	-48.450	-29.221	-28.904	-34.286	-14.602	-41.470	-12.048
	32	$[2, 3, 7, 4, 2, 1, 1, 2, 0, 3, 3, 4]^T$	-108.294	-36.742	-44.546	-40.673	-10.891	-43.230	-11.828
	33	$[6, 2, 3, 3, 0, 1, 2, 2, 4, 2, 5, 3]^T$	-18.993	-24.720	-40.930	-21.034	-44.568	-39.799	-11.141
	35	$[4, 2, 3, 5, 2, 4, 3, 2, 1, 0, 6, 3]^T$	-133.796	-41.694	-41.537	-38.038	-33.786	-45.000	-26.719
LD	27	$[0, 2, 7, 7, 0, 0, 0, 0, 0, 0, 11, 0]^T$	-122.392	-37.126	-36.830	-9.354	-22.173	-37.233	-29.108
	28	$[0, 4, 6, 5, 1, 0, 0, 1, 0, 0, 9, 2]^T$	-130.506	-37.941	-37.814	-31.011	-24.183	-38.452	-30.382
	30	$[0, 3, 4, 8, 3, 0, 1, 0, 0, 0, 6, 5]^T$	-149.790	-42.406	-42.493	-26.566	-36.119	-42.315	-35.032
	30	$[0, 3, 5, 6, 1, 0, 2, 1, 1, 0, 3, 8]^T$	-142.861	-38.852	-39.129	-32.708	-38.979	-38.529	-30.878
	30	$[0, 3, 5, 6, 1, 0, 2, 1, 1, 0, 3, 8]^T$	-141.862	-38.576	-38.042	-31.762	-38.070	-38.543	-32.338

TABLE II: Comparison between the Elephant Herding Optimization method proposed by Horng and Lin (2020) [27] and our adaptation for the Lagrangian duality approach of Pot et al. (2008) [25]. Minimum cost staffing per hour of execution time of the best run of Table I.

Time (h)	**Method**	**Cost**	**Staffing**	$\text{SL}_1 - \tau_1$	$\text{SL}_2 - \tau_2$	$\text{SL}_3 - \tau_3$	$\text{SL}_4 - \tau_4$	$\text{SL}_5 - \tau_5$	$\text{SL}_6 - \tau_6$	$\text{SL}_7 - \tau_7$
1	EHO	27	$[3, 3, 6, 4, 1, 0, 2, 0, 0, 1, 5, 2]^T$	-17.110	-30.564	-37.099	-2.229	0.260	-32.116	-2.520
2	EHO	26	$[2, 4, 3, 3, 2, 0, 0, 1, 0, 0, 5, 6]^T$	-26.284	-17.138	-16.915	-15.393	-0.689	-15.689	-22.571
	LD	24	$[2, 3, 2, 3, 3, 0, 2, 0, 2, 1, 4, 2]^T$	3287.789	-22.775	-33.291	10.695	-35.095	-21.528	24.750
3	EHO	25	$[0, 4, 4, 3, 1, 0, 0, 1, 1, 0, 4, 7]^T$	140.776	-2.520	-3.530	-9.261	-23.243	-2.504	-14.823
	LD	24	$[2, 3, 2, 3, 3, 0, 2, 0, 2, 1, 4, 2]^T$	3287.789	-22.775	-33.291	10.695	-35.095	-21.528	24.750
4	EHO	26	$[2, 5, 5, 5, 2, 1, 1, 1, 2, 0, 3, 4]^T$	-23.450	-15.180	-15.933	-13.795	-29.171	-15.320	-20.839
	LD	30	$[4, 1, 8, 4, 1, 0, 0, 0, 1, 0, 8, 3]^T$	-145.445	-40.157	-39.573	-31.102	-29.959	-41.403	-34.485

TABLE III: Average staffing employed by the firm and its performance w.r.t. the SL constraints.

Cost	**Staffing**	$\text{SL}_1 - \tau_1$	$\text{SL}_2 - \tau_2$	$\text{SL}_3 - \tau_3$	$\text{SL}_4 - \tau_4$	$\text{SL}_5 - \tau_5$	$\text{SL}_6 - \tau_6$	$\text{SL}_7 - \tau_7$
63	$[14, 11, 8, 9, 6, 0, 3, 0, 1, 0, 2, 9]^T$	-160.000	-45.000	-45.000	-45.000	-45.000	-45.000	-44.942

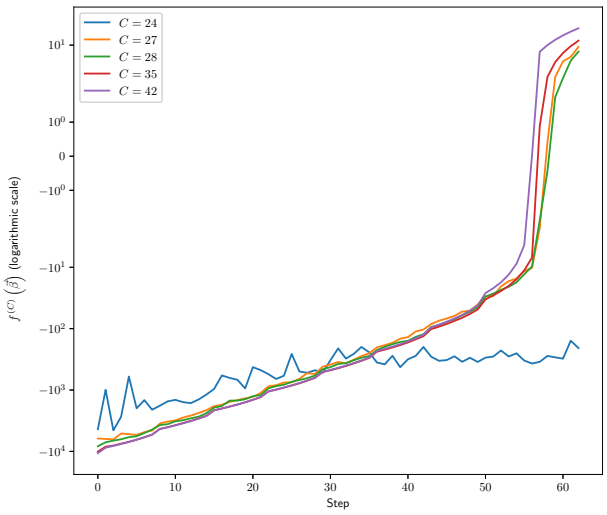


Fig. 5: Plot of $f^{(C)}\left(\vec{\beta}\right)$ as the algorithm adjusts $\vec{\beta}$ for a given C. Ever step is an adjustment of $\vec{\beta}$.

would require either longer simulation times or further work regarding performance approximation.

VIII. Conclusions

In this paper, we have demonstrated how to model a multi-skill priority routing call center taking call data as a starting point. We have assessed the quality of our model by means of the Kolmogorov-Smirnov test and by comparing simulation results to real observations.

Then, an overview of the existing research regarding call centers was presented, paying special attention to papers on staffing multi-skill call centers, which were treated in a one-by-one basis in chronological order.

Once the problem and context were adequately defined, an existing approach to staffing based on Lagrangian duality was adapted for a use case with multiple service level constraints. Our algorithm was tested against the latest publication, which employs a population-based evolutionary algorithm inspired by elephant herds. It was found that, while the elephant-based approach is 4 times faster than our method, we were able to produce minimal cost staffings with lower variance and similar quality to our quick competitor. Moreover, possible improvements to both algorithms were pointed out.

Finally, we compared the output of EHO and LD to the staffing which the firm is currently employing. At first glance, there is a huge difference between the costs of the solutions provided by the algorithms and the cost of the current agent composition. However, we must be careful treating such *virtual* staffings, because agents do not work non-stop in real life. How to translate the output of the algorithm into reality requires further investigation.

Acknowledgment

The author would like to thank Dr. Matúš Mihalák for being the main source of supervision, guidance and feedback. The author is also grateful for the help of Dr. Joël Karel, who provided the initial code for the queuing system implementation. Last but not least, this paper wouldn't be possible without the essential collaboration and contributions of some employees of the firm.

References

[1] "The History of the Call Centre – Updated," *Call Centre Helper Magazine*, oct 2019. [Online]. Available: https://www.callcentrehelper.com/the-history-of-the-call-centre-15085.htm

[2] M. Shacklett, "How artificial intelligence is taking call centers to the next level," *TechRepublic*. [Online]. Available: https://www.techrepublic.com/article/how-artificial-intelligence-is-taking-call-centers-to-the-next-level/

[3] E. Mazareanu, "Market size of the call center market worldwide from 2012 to 2017, by region," in *Statista - The Statistics Portal*, oct 2019. [Online]. Available: https://www.statista.com/statistics/881033/call-center-market-size-region/

[4] Grand View Research. (2020, feb) Contact Center Software Market Size Worth $72.3 Billion By 2027. [Online]. Available: https://www.grandviewresearch.com/press-release/global-contact-center-software-market

[5] A. Brandt and M. Brandt, "On a two-queue priority system with impatience and its application to a call center," *Methodology and Computing in Applied Probability*, vol. 1, no. 2, pp. 191–210, 1999.

[6] O. Garnett, A. Mandelbaum, and M. Reiman, "Designing a call center with impatient customers," *Manufacturing & Service Operations Management*, vol. 4, no. 3, pp. 208–227, 2002.

[7] Y.-J. ZHU and R.-X. ZHU, "Performance analysis of call centers based on m/m/s/k+ m queue with retrial and impatience [j]," *Journal of Jiangsu University (National Science Edition)*, vol. 5, 2004.

[8] J. R. Artalejo and V. Pla, "On the impact of customer balking, impatience and retrials in telecommunication systems," *Computers & Mathematics with Applications*, vol. 57, no. 2, pp. 217–229, 2009.

[9] O. Jouini, Y. Dallery, and Z. Akşin, "Queueing models for full-flexible multi-class call centers with real-time anticipated delays," *International Journal of Production Economics*, vol. 120, no. 2, pp. 389–399, 2009.

[10] O. Jouini and A. Roubos, "On multiple priority multi-server queues with impatience," *Journal of the Operational Research Society*, vol. 65, no. 5, pp. 616–632, 2014.

[11] A. D. Ridley, M. C. Fu, and W. A. Massey, "Fluid approximations for a priority call center with time-varying arrivals," in *Winter Simulation Conference*, vol. 2, 2003, pp. 1817–1823.

[12] R. A. Shumsky, "Approximation and analysis of a call center with flexible and specialized servers," *OR Spectrum*, vol. 26, no. 3, pp. 307–330, 2004.

[13] F. Iravani and B. Balcıoğlu, "Approximations for the m/gi/n+gi type call center," *Queueing Systems*, vol. 58, no. 2, pp. 137–153, 2008.

[14] G. Koole and A. Mandelbaum, "Queueing models of call centers: An introduction," *Annals of Operations Research*, vol. 113, no. 1-4, pp. 41–59, 2002.

[15] G. Koole and A. Pot, "An overview of routing and staffing algorithms in multi-skill customer contact centers," 2006.

[16] S. Bhulai, "Dynamic routing policies for multiskill call centers," *Probability in the Engineering and Informational Sciences*, vol. 23, no. 1, p. 101, 2009.

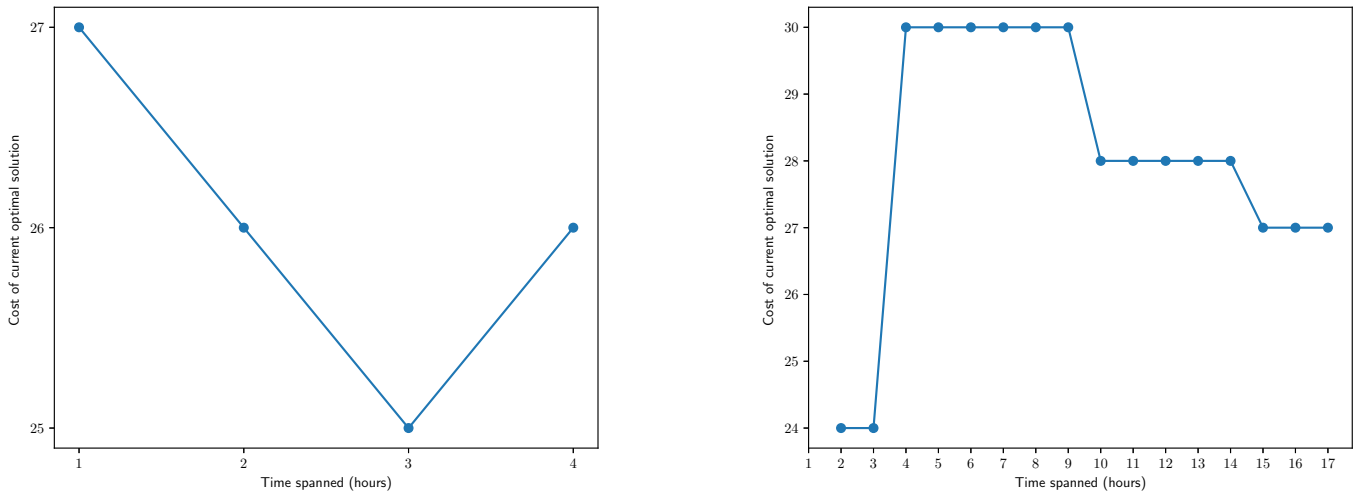
[17] W. Chan, G. Koole, and P. l'Ecuyer, "Dynamic call center routing policies using call waiting and agent idle times," *Manufacturing & Service Operations Management*, vol. 16, no. 4, pp. 544–560, 2014.

[18] J. M. Harrison and A. Zeevi, "Dynamic scheduling of a multiclass queue in the halfin-whitt heavy traffic regime," *Operations Research*, vol. 52, no. 2, pp. 243–257, 2004.

[19] A. N. Avramidis, M. Gendreau, P. L'Ecuyer, and O. Pisacane, "Simulation-based optimization of agent scheduling in multiskill call centers." 2007.

[20] J. Atlason, M. A. Epelman, and S. G. Henderson, "Optimizing call center staffing using simulation and analytic center cutting-plane methods," *Management Science*, vol. 54, no. 2, pp. 295–309, 2008.

[21] A. N. Avramidis, W. Chan, M. Gendreau, P. L'ecuyer, and O. Pisacane, "Optimizing daily agent scheduling in a multiskill call center," *European Journal of Operational Research*, vol. 200, no. 3, pp. 822–832, 2010.



(a) Elephant Herding Optimization

(b) Lagrangian duality

Fig. 6: Minimum cost staffing per hour of execution time. Best run of Table I.

[22] S. Mattia, F. Rossi, M. Servilio, and S. Smriglio, "Staffing and scheduling flexible call centers by two-stage robust optimization," *Omega*, vol. 72, pp. 25–37, 2017.

[23] A. N. Avramidis and P. L'Ecuyer, "Modeling and simulation of call centers," in *Proceedings of the Winter Simulation Conference, 2005.* IEEE, 2005, pp. 9–pp.

[24] 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 & Service Operations Management*, vol. 17, no. 4, pp. 571–588, 2015.

[25] A. Pot, S. Bhulai, and G. Koole, "A simple staffing method for multiskill call centers," *Manufacturing & service operations management*, vol. 10, no. 3, pp. 421–428, 2008.

[26] A. N. Avramidis, W. Chan, and P. L'Ecuyer, "Staffing multi-skill call centers via search methods and a performance approximation," *Iie Transactions*, vol. 41, no. 6, pp. 483–497, 2009.

[27] S.-C. Horng and S.-S. Lin, "Staffing optimization of one-period multi-skill call center," in *Proceedings of the 2020 the 3rd International Conference on Computers in Management and Business*, 2020, pp. 89–93.

[28] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[29] Scipy. (2020) Numpy and Scipy Documentation. [Online]. Available: https://docs.scipy.org/doc/

[30] I. M. Chakravarty, J. Roy, and R. G. Laha, "Handbook of methods of applied statistics," 1967.

[31] J. M. Harrison and A. Zeevi, "A method for staffing large call centers based on stochastic fluid models," *Manufacturing & Service Operations Management*, vol. 7, no. 1, pp. 20–36, 2005.

[32] R. B. Wallace and W. Whitt, "A staffing algorithm for call centers with skill-based routing," *Manufacturing & Service Operations Management*, vol. 7, no. 4, pp. 276–294, 2005.

[33] M. T. Cezik and P. L'Ecuyer, "Staffing multiskill call centers via linear programming and simulation," *Management Science*, vol. 54, no. 2, pp. 310–323, 2008.

[34] J. E. Kelley, Jr, "The cutting-plane method for solving convex programs," *Journal of the society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[35] J. Atlason, M. A. Epelman, and S. G. Henderson, "Call center staffing with simulation and cutting plane methods," *Annals of Operations Research*, vol. 127, no. 1-4, pp. 333–358, 2004.

[36] P. Chevalier and J.-C. Van den Schrieck, "Optimizing the staffing and routing of small-size hierarchical call centers," *Production and Operations Management*, vol. 17, no. 3, pp. 306–319, 2008.

[37] A. Fredericks, "Congestion in blocking systems—a simple approximation technique," *Bell System Technical Journal*, vol. 59, no. 6, pp. 805–827, 1980.

[38] N. Tabordon, "Modeling and optimizing the management of operator training in a call center," Ph.D. dissertation, UCL-Université Catholique de Louvain, 2002.

[39] G. Koole and J. Talim, "Exponential approximation of multi-skill call centers architecture," *Proceedings of QNETs*, vol. 23, pp. 1–10, 2000.

[40] W. Chan, T. A. Ta, P. l'Ecuyer, and F. Bastin, "Two-stage chance-constrained staffing with agent recourse for multi-skill call centers," in *2016 Winter Simulation Conference (WSC)*. IEEE, 2016, pp. 3189–3200.

[41] T. A. Ta, W. Chan, F. Bastin, and P. L'Ecuyer, "A simulation-based decomposition approach for two-stage staffing optimization in call centers under arrival rate uncertainty," *Submitted for publication*, 2019.

[42] G.-G. Wang, S. Deb, X.-Z. Gao, and L. D. S. Coelho, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour," *International Journal of Bio-Inspired Computation*, vol. 8, no. 6, pp. 394–409, 2016.

[43] A. E. Hassanien, M. Kilany, E. H. Houssein, and H. AlQaheri, "Intelligent human emotion recognition based on elephant herding optimization tuned support vector regression," *Biomedical Signal Processing and Control*, vol. 45, pp. 182–191, 2018.