An Exact Optimization Approach for Personnel Scheduling Problems in the Call Center Industry

Rita Martins[1], Telmo Pinto[1,2(✉)], and Cláudio Alves[1]

[1] Centro ALGORITMI, University of Minho, Braga, Portugal
claudio@dps.uminho.pt
[2] CEMMPRE, University of Coimbra, Coimbra, Portugal
telmo.pinto@dem.uc.pt

Abstract. Nowadays, the importance of the call center industry is increasing because they are a major means of communication between organizations and their customers. So, ensuring optimized personnel schedules in call centers has several advantages, such as reduced total labor costs, overstaffing, increased employee satisfaction, and acceptable waiting times. In this paper, we address the personnel scheduling problem in a 24/7 call center where the scheduling process is done manually. So, the main goal is to explore exact solution approaches to achieve better solutions while reducing the processing time. The proposed optimization model is an Integer Programming model to assign shifts to workers while minimizing the total penalization associated with employees' time preferences. The model is tested with several instances, including randomly generated and real-world data instances. The quality of the model is assessed through a computational study of its linear relaxation, concluding that the model presents null integrality gaps in all the tested instances. Additionally, to evaluate the performance of the model when running large instances, several randomly generated instances were tested, achieving good computational results.

Keywords: personnel scheduling · call centers · optimization · integer programming

1 Introduction

Over the last few years, the importance of studying personnel scheduling problems increased considerably. This is due to the impact of the labor cost on companies' costs [7]. Therefore, reducing this cost, even just a little, can be very advantageous and worthwhile. According to Bard et al. [4], weak personnel schedules can lead to an oversupply of workers with too much idle time, or an undersupply with an attendant loss of business. Thompson [18] reports three reasons for paying particular attention to staff scheduling. The first consists of the employee's preferences since a work schedule that comes reasonably close

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14105, pp. 407–424, 2023.
https://doi.org/10.1007/978-3-031-37108-0_26

to meeting them can improve customer service. The second reason is related to the real-time spent on developing a labor schedule, which can leave less time for the scheduler to manage the employees and interact with customers. Finally, the third reason is profitability and effectiveness since short-term overstaffing and long-term understaffing can be reduced.

One of the main goals of an organization is to satisfy the customer and also the employees in a cost-effective manner. There are many concerns to consider, such as shift equity, staff preferences, flexible workplace agreements, and part-time work. Thus, decision support systems must be carefully implemented to achieve optimized staff schedules. Personnel scheduling consists of assigning workers shifts and days off so that each has a line of work. This process can be complex because schedulers must assign employees to shifts and tasks while accounting for different rules and regulations and workers' availability. In the first phase of this process, it is necessary to have an idea of the number of staff needed to meet the service demand and also to know the particular skills of all individuals so that they can be allocated to shifts whose tasks meet their skills.

As time goes by, the relative importance of satisfying the needs of employees in staffing and scheduling decisions has increased. Nowadays, organizations want to offer some conditions to their employees that were not a concern in the past. Consequently, and as referred to above, organizations create work schedules considering employee preferences [5] and trying to offer part-time contracts or flexible work hours [4]. Mohan [15] considers a staff scheduling problem consisting entirely of part-time workers and also considers their shifts' preferences.

Several personnel scheduling problems with different characteristics can be found in the literature. This fact can be explained since, as can be expected, different work environments imply different requirements. Therefore, some features receive particular attention, such as workforce characteristics, shift flexibility, the type of constraints and objective function used to model the problem, and the selection of the solution method to solve the problem. These features will be presented below.

Several classification methods for personnel scheduling problems can be found, providing a general framework for classifying the contributions carried out in this area. Baker [3] proposed the first one, which is commonly studied in the literature for general labor scheduling [6].

Van den Bergh et al. [7] proposed a classification method that organizes the literature review using different perspectives. Therefore, the classification fields are the following: personnel characteristics, decision delineation, shift definition; constraints, performance measures, and flexibility; solution method and uncertainty incorporation; and application area and applicability of research. Each field is described in detail as follows.

1.1 Personnel Characteristics, Decision Delineation and Shifts Definition

In order to classify personnel members, one can look at their labor contracts. In this way, it is possible to distinguish two categories: full-time and part-time

workers. Mohan [15] considered a scheduling problem where the workforce is composed entirely of part-time workers, claiming that more people are opting to work part-time due to the flexibility it offers. Bard et al. [4] considered both full-time and part-time workers. In some cases, casual workers, such as workers from another company division, are also considered to deal with the lack of a regular workforce. Another existing classification is based on the grouping of employees, where the workers are categorized into individual or crew entities. Usually, this last one can be found in scheduling problems in the transportation area. Besides that, one must remember that different employees have different availability and also different individual personal preferences, which are both important since one of the main goals of an organization is to satisfy the employees.

According to Felici and Gentile [13], a shift is a work duty characterized by a starting time and a duration. Shifts can be fixed or variable in terms of starting-time or length for each employee and each day. In personnel shift scheduling, shifts and days-off are assigned to each worker so that every employee has a line of work while considering some existing conditions or requirements.

As expected, in personnel scheduling problems, making some decisions is imperative and inevitable. The main ones are related to the assignment of tasks (e.g., a given employee is assigned to a certain task) and shift sequences (e.g., a given employee works on day 1 in the morning, but he/she does not work on day 2 and works on day 3 in the morning). The task assignment methods depend on whether tasks times are fixed or movable, breaks exist in shifts, overtime is allowed, or specific skills or qualifications are required to perform certain tasks. According to Smet et al. [17], tasks should be incorporated in the construction of rosters for employees to reduce operational expenses while preserving a high quality of service. Maenhout and Vanhoucke [14] claimed that when the scheduler integrates personnel shift scheduling and personnel task scheduling, the obtained personnel roster is improved.

1.2 Application Area: Scheduling in Call Centers

According to Ernst et al. [11], staff scheduling methods have been applied to transportation systems such as airlines and railways, health care systems, emergency services such as police, ambulance, and fire brigade, call centers, and many other service organizations such as hotels, restaurants, and retail stores. Van den Bergh et al. [7] add other application areas like military, supermarkets, festivals, and parking lots personnel scheduling problems. Even so, there are two applications of staff scheduling and rostering that stand out from the other: transportation services, particularly airline crew scheduling, and rostering because of its economic scale and impact, and nurse scheduling since it is unacceptable not to fully support patient care needs.

Since the case study to be considered concerns personnel scheduling in a call center environment, a brief literature review of this application area will be presented.

The importance of call centers for companies is widely increasing because, although they are not the only ones, they are a vital channel for companies

to communicate with their customers [12], having a crucial role in customer acquisition and retention [1]. Consequently, having a good scheduling system is fundamental because an inadequate number of agents for each shift affects customer satisfaction, for instance, because of long waiting times and labor-related costs. According to Cordone et al. [10], a major part of the call center's operational costs are related to labor costs. That is why the staff scheduling problem became the main problem in call center operations. However, it is not easy to solve due to the uncertainty of the number of tasks that are needed to be performed.

According to Robbins and Harrison [16], some OR methodologies, such as queuing theory, optimization, and simulation, can be found in the literature on call centers' problems. Usually, solving shift scheduling problems in a call center can be divided into two phases. In the first one, the goal is to determine the number of required agents for each period, considering some customer service measures like customer waiting times or abandonment rates. For this task, some authors use queuing and simulation models to obtain the correct numbers, as can be seen in [12] and [9]. The second phase is defined by the scheduling algorithm, where the goal is to allocate employees to the shifts, considering a particular objective function, as seen before.

Bhulai, Koole, and Pot [8] propose an integer programming model for the shift scheduling problem in a call center, where they considered a multiskill workforce, and their objective is to obtain a good service-level against minimal costs. In this sense, the presented method is divided into two phases: determination of the required number of agents for each period, considering service-level constraints, and determination of an optimal set of shifts, minimizing costs while satisfying the required workforce calculated in the first step. The service-level constraints studied include a minimum bound of calls with a waiting time of less than 20 s. Considering a workforce with two distinct skills, shifts of five and six hours, and satisfying the service-level constraint for 80% of the calls, the optimal number of shifts is 167. The authors checked the optimality of the methods from both steps and concluded that the solution obtained is less than 3% from the optimal objective value. In conclusion, the authors highlighted the short computation time required and easy implementation, as also referred that, although the method was developed for call centers, it can be used for other service systems that involve multiple distinguished skills, agents working on different tasks, and employees with identical productivity within the same skill group.

Alfares [2] presented a two-step solution method to minimize the labor cost in a call center by determining the optimum number of agents and their schedules to meet demand. In the first step, queuing theory converts extensive data into demand values for a typical workweek. Then, an integer programming model is built to construct detailed weekly schedules. These schedules include each agent's work hours, meal breaks, and off days. The problem approached considers a call center that operates 24 h a day, 7 days of week. The 47 agents working there are divided into three groups, having different pay scales and work schedules. There are two problems in this organization to be tackled: customers are not completely

satisfied because they are placed on hold for several minutes during peak periods, and the 1-hour breaks the workers are entitled to are unscheduled. Before the study of the solution method proposed, using no scientific methods but based on the management's observations, a schedule for each worker was built. Then, two instances were solved using LINDO optimization software, differing in the minimum number of employees required in the considered hours. The solutions obtained and the *a priori* solution were compared. Although both of the solutions given by the integer programming model were better than the *a priori* schedule, since the workforce size is reduced, the workforce utilization is increased, and the labor cost decreased, the management chose one of them because it was the one that minimizes most the waiting time.

The purpose of the work presented in this paper is to create timetables faster and with higher quality than those currently created manually, using exact approaches to solve personnel scheduling issues at call centers. This paper is organized as follows. Section 2 concerns the problem's main characteristics and the description of an example of an instance. In Sect. 3, the developed Integer Programming model is presented and described. In Sect. 4, the computational experiments are described and discussed. Finally, Sect. 5 summarizes the principal findings, resulting conclusions, and suggestions for future work.

2 Problem Definition

The call center considered in this work operates 24 h a day and 7 days a week, having a new shift starting every 30 min (the company discretizes the demand of the workforce for each interval of 30 min). This means that, in a day, there are 48 different shifts: the first shift starts at 00 h 00, and the last shift starts at 23 h 30. Each shift has a fixed duration of 9 h and some mandatory breaks that are informally managed within the team. The planning horizon of the problem is one month, divided into periods of 30 min that constitute set H. So, for example, considering a planning horizon of a single day, set H is formed by 48 elements, as can be seen in Fig. 1.

hours	00:00:00	00:30:00	01:00:00	01:30:00	02:00:00	02:30:00	03:00:00	03:30:00	04:00:00	04:30:00	05:00:00	05:30:00
H	0	1	2	3	4	5	6	7	8	9	10	11
hours	06:00:00	06:30:00	07:00:00	07:30:00	08:00:00	08:30:00	09:00:00	09:30:00	10:00:00	10:30:00	11:00:00	11:30:00
H	12	13	14	15	16	17	18	19	20	21	22	23
hours	12:00:00	12:30:00	13:00:00	13:30:00	14:00:00	14:30:00	15:00:00	15:30:00	16:00:00	16:30:00	17:00:00	17:30:00
H	24	25	26	27	28	29	30	31	32	33	34	35
hours	18:00:00	18:30:00	19:00:00	19:30:00	20:00:00	20:30:00	21:00:00	21:30:00	22:00:00	22:30:00	23:00:00	23:30:00
H	36	37	38	39	40	41	42	43	44	45	46	47

Fig. 1. Example: set of time periods with a planning horizon of a day

Each worker cannot work more than 5 consecutive working days every 7 days and can only be on one shift per day. Additionally, between two consecutive shifts, each worker must stop for at least 11 h. Ideally, one worker has the same shift for a maximum of 2 weeks, creating a stable route for each one to provide a comfortable schedule for the workers. However, the company intended a maximum of 2 weeks since shifts could be changed to ensure the same workers were not assigned to the same shifts considered less attractive.

The workforce is divided geographically between two industrial centers, where the workers are also divided into teams according to their language and team skills. This means that each worker w will have a corresponding skill set K_w, composed by the different teams that he/she can be assigned because he/she has the required skills. To guarantee workers' satisfaction, their time preferences are considered using a penalization system: the exact preferred time has a minimum penalization associated, *i.e.* 1, which increases exponentially as the considered time gets away from the preferred time.

The scheduling goal is to assign workers to teams and shifts, minimizing the total penalization when considering workers' preferences and ensuring that the workforce requirements r^{kh} for certain team k at a certain time period h are covered. In this organization, a monthly personnel scheduling is carried out, and this process is done manually by a scheduler. So, the current process time is larger than expected, having a duration of some days. Figures 2 and 3 represent the schedules obtained by this process referring to two different teams of the considered call center.

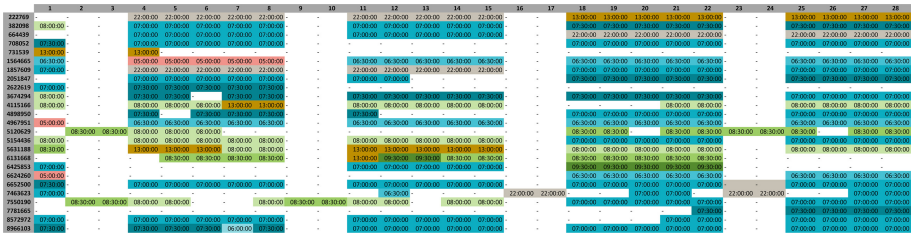

Fig. 2. Schedule of team 1 obtained manually

The following notation is used in the development of the model:

Indices and Sets

k: index for the set of teams K;
h: index for the set of time periods H;
w: index for the set of workers W;
H: set of time periods;
K: set of teams;
W: set of workers

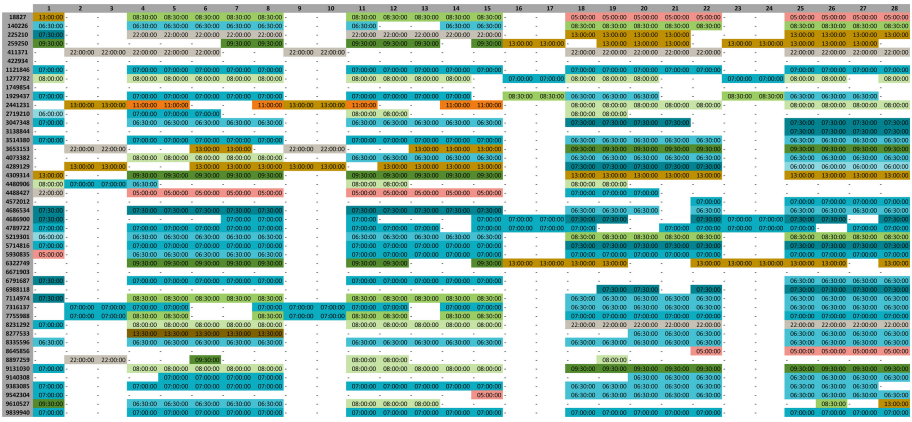

Fig. 3. Schedule of team 2 obtained manually

Parameters

K_w: set of teams to which a worker w ($w \in W$) can be assigned because he/she has the required skills;
p_w^h: penalization of assigning worker w on time period h;
r^{kh}: number of workers that are needed at team k and time period h;

2.1 Example of an Instance

An instance of this problem is defined based on the notation presented in the previous section. In this section, an example of an instance will be described.

Consider a scenario where the planning horizon is one day. There is only 1 working team composed of 3 different workers. Table 1 shows the data referring to four parameters: teams, time periods, workers, and the sets associated with each worker that gives the list of teams where he/she can work in. Note that all the workers are identified by an ID number composed only of integer numbers (Table 2).

Table 1. Instance example: teams, time periods and workers

$K = \{1\}$
$H = \{0, ..., 47\}$
$W = \{1, 2, 3\}$

In this particular case, only one worker is required to work the shift starting at 8 h 00. Concerning the preference's penalizations, one has three different situations: worker 1 prefers to work on 17 h 30 shift, which means that, in that hour, the associated penalization will be minimum (*i.e.*, equals to 1); worker 2 prefers to work on 07 h 00 shift, which means that a penalization of 1 is associated to

Table 2. Instance example: set of teams to which a worker w can be assigned

w	K_w
1	{1}
2	{1}
3	{1}

that hour; and worker 3 prefers to work on 14 h 30 shift, which means that, in that hour, the associated penalization will also be minimum. The considered values of both necessities and preference's penalizations can be seen in Table 3 and Table 4.

Table 3. Instance example: workforce requirements

period time	team 1	period time	team 1	period time	team 1
0	0	16	1	32	1
1	0	17	1	33	1
2	0	18	1	34	0
3	0	19	1	35	0
4	0	20	1	36	0
5	0	21	1	37	0
6	0	22	1	38	0
7	0	23	1	39	0
8	0	24	1	40	0
9	0	25	1	41	0
10	0	26	1	42	0
11	0	27	1	43	0
12	0	28	1	44	0
13	0	29	1	45	0
14	0	30	1	46	0
15	0	31	1	47	0

3 An Integer Programming Model

This work aims to develop an efficient exact solution method to solve personnel scheduling problems at call centers providing automatic schedules instead of manual ones. Thus, considering the problem described in Sect. 2, the following integer programming (IP) model was developed.

Decision Variables

$$x_w^{kh} = \begin{cases} 1, \text{ if worker } w \text{ is assigned to team } k, \text{ and to the shift } h, \\ 0, \text{ otherwise.} \end{cases} \quad (1)$$

Table 4. Instance example: preferences' penalizations

period time	worker 1	worker 2	worker 3
0	64	128	512
1	64	64	512
2	128	64	1024
3	128	32	1024
4	256	32	2048
5	256	16	2048
6	512	16	2048
7	512	8	1024
8	1024	8	1024
9	1024	4	512
10	2048	4	512
11	2048	2	256
12	2048	2	256
13	1024	1	128
14	1024	1	128
15	512	1	64
16	512	1	64
17	256	1	32
18	256	2	32
19	128	2	16
20	128	4	16
21	64	4	8
22	64	8	8
23	32	8	4
24	32	16	4
25	16	16	2
26	16	32	2
27	8	32	1
28	8	64	1
29	4	64	1
30	4	128	1
31	2	128	1
32	2	256	2
33	1	256	2
34	1	512	4
35	1	512	4
36	1	1024	8
37	1	1024	8
38	2	2048	16
39	2	2048	16
40	4	2048	32
41	4	1024	32
42	8	1024	64
43	8	512	64
44	16	512	128
45	16	256	128
46	32	256	256
47	32	128	256

IP Model

$$\min. \sum_{w \in W} \sum_{k \in K_w} \sum_{h \in H} p_w^h x_w^{kh} \tag{2}$$

s.t.

$$\sum_{h=i}^{min(|H|-1,i+335)} \sum_{k \in K_w} x_w^{kh} \leq 5, \ \forall w \in W, \ \forall i \in I = \{h \in H : h \bmod 336 = 0\}, \tag{3}$$

$$\sum_{h=f}^{f+47} \sum_{k \in K_w} x_w^{kh} \leq 1, \ \forall w \in W, \ \forall f \in F = \{h \in H : h \bmod 48 = 0\}, \tag{4}$$

$$\sum_{\{w \in W : k \in K_w\}} \sum_{h=\max\{0,b-17\}}^{b} x_w^{kh} \geq r^{kb}, \ b \in H, \ \forall k \in K, \tag{5}$$

$$\sum_{k \in K_w} \sum_{h=i}^{i+39} x_w^{kh} \leq 1, \forall w \in W, i \in J = \{h \in H : h \leq |H| - 40\}. \tag{6}$$

$$x_w^{kh} \in \{0,1\}, \forall w \in W, \forall k \in K_w, \forall h \in H \tag{7}$$

The purpose of this IP model is to assign workers to teams and shifts. To formulate the problem, a set of decision variables (1) is used to indicate if a worker is assigned to a certain team and to the shift starting at a certain period of time. The objective function (2) is to minimize the total penalization when considering workers' preferences.

Constraints (3) and (4) ensure that each worker cannot work more than 5 days per week and can only be in one shift per working day, respectively. In (3), the set $I = \{h \in H : h \bmod 336 = 0\}$ is composed by all the time periods $h \in H$ that are divisors of 336. In practice, all the time periods that belong to this set are the first period of time of every 7-day week because 336 time periods of 30 min correspond to a whole week, starting at 00 h 00 of day 1 to 23 h 30 of day 7. So, to ensure that each worker only works a maximum of 5 days per week, the set I guarantees that the summed shifts correspond to only a week. Additionally, the upper bound of the summation in I is given by $min(|H|-1, i+335)$, where $i \in I$, because of the possibility of the month finishing in the middle of a week. If the considered month does not include the whole week, then the upper bound of the summation will be $|H|-1$. Otherwise, the upper bound will be $i+335$, where $i \in I$. In (4), the logic is the same, but the time horizon is only a day, that is, 48 time periods of 30 min. Therefore, the set $F = \{h \in H : h \bmod 48 = 0\}$ is composed by all the time periods $h \in H$ that are divisors of 48, that is, all the time periods that belong to this set are the first period of time of every day. In this case, the upper bound of the summation in F is only given by $f+47$, where $f \in F$, because all the days considered are complete.

Constraints (5) force the number of allocated workers to a certain shift in a certain period of time and team to be at least equal to the number of required workforce in that same period of time and team. In this constraint, the lower and upper bounds of the summation in H are, respectively, $\max\{0, b-17\}$ and b, where $b \in H$. This is because, in order to know if the required number of

workers in a certain time period is covered, it is necessary to sum all the workers that started working a maximum of 08 h 30, *i.e.* 17 time periods, before the time period considered until the ones that start working in the exact time period considered. In the beginning of every month, until the considered time period is less than 17, the sum is done from the time period 0, not considering the data of the previous month.

Constraints (5) force the number of allocated workers to a certain shift in a certain period of time and team to be at least equal to the number of required workforce in that same period of time and team. In this constraint, the lower and upper bounds of the summation in H are, respectively, $\max\{0, b-17\}$ and b, where $b \in H$. This is because, to know if the required number of workers in a certain time period is covered, it is necessary to sum all the workers that started working a maximum of 08 h 30, *i.e.* 17 time periods, before the time period considered until the ones that start working in the exact time period considered. At the beginning of every month, until the considered time period is less than 17, the sum is done from the time period 0, not considering the data of the previous month.

Constraints (6) establish, for each worker, a mandatory break of at least 11 h between two consecutive shifts. To ensure that, a time interval of 40 time periods is considered. This interval comprises 18 time periods corresponding to a 9-hour shift plus 22 time periods corresponding to an 11-hour break. To deal with the end of the month, the set $J = \{h \in H : h \leq |H| - 40\}$ is used, guaranteeing that the limit is not overpast.

To include workers' preferences, and based on the quantification of preference violations proposed by Bard and Purnomo [5], a penalization system was built. Let p be the time preference of a worker. Considering that $h \in \{0, ..., 47\}$ represents the $(h+1)$th 30-min time period of a day, a penalty point v is fixed to calculate a penalization value 2^{v-1}, according to the difference between h and p, as seen in Table 5.

Table 5. Penalization System

violation	penalty points v	penalization 2^{v-1}
$0 < \|h-p\| \leq 2 \vee \|h-p\| > 45$	1	1
$2 < \|h-p\| \leq 4 \vee 43 < \|h-p\| \leq 45$	2	2
$4 < \|h-p\| \leq 6 \vee 41 < \|h-p\| \leq 43$	3	4
$6 < \|h-p\| \leq 8 \vee 39 < \|h-p\| \leq 41$	4	8
$8 < \|h-p\| \leq 10 \vee 37 < \|h-p\| \leq 39$	5	16
$10 < \|h-p\| \leq 12 \vee 35 < \|h-p\| \leq 37$	6	32
$12 < \|h-p\| \leq 14 \vee 33 < \|h-p\| \leq 35$	7	64
$14 < \|h-p\| \leq 16 \vee 31 < \|h-p\| \leq 33$	8	128
$16 < \|h-p\| \leq 18 \vee 29 < \|h-p\| \leq 31$	9	256
$18 < \|h-p\| \leq 20 \vee 27 < \|h-p\| \leq 29$	10	512
$20 < \|h-p\| \leq 22 \vee 25 < \|h-p\| \leq 27$	11	1024
$22 < \|h-p\| \leq 25$	12	2048

4 Computational Experiments and Analysis

Several computational experiments were executed to evaluate and analyze the performance of the model. The code is implemented in IBM ILOG CPLEX Optimization Studio 12.7.0.0, and the tests are carried out on an Intel Core i7-6500U processor and 8 GB of memory RAM, with Windows 10 operating system.

A way to obtain information about the IP problem is through the solution of its linear relaxation. The linear programming (LP) relaxation of an integer programming model is the problem that arises when the integrality constraints are relaxed, which means that all variables are allowed to take non-integer values. Hence, the optimal solution to the LP is not necessarily integer. However, since the feasible region of the LP is larger than the feasible region of the IP, the optimal value of the former is no worse than the optimal value of the latter. This implies that the optimal value to the LP is a lower bound on the optimal value for the problem we started with. The gap between the optimal LP value and the optimal integral solution is called the integrality gap of the linear program, and it is given by

$$IG = \frac{OPT_{int}}{OPT_{frac}}, \tag{8}$$

where OPT_{int} is the optimal value of the IP and OPT_{frac} is the optimal value of the LP.

In order to analyze the integrality gap of the linear program and to evaluate the quality of the IP model, a set of 55 randomly generated instances and a set of 3 real-data instances were tested, in which the planning horizon is a 28-day month or a 31-day month. While the number of workers in the first set varies between 30, 50, 75, 100, and 150, in the second set it only varies between 55 and 71 workers. More information about which one of the instances, including the number of teams, number of workers required, and values of penalizations are described in Table 6.

The results are given in Table 7. Column OPT_{int} and OPT_{frac} give the optimal value of the IP model and the optimal value of the LP relaxation, respectively. Assuming the same thinking, column CPU_{int} time gives the processing time (minutes) for each one of the instances executed with the IP model, and column CPU_{frac} time the processing time (minutes) for each one of the instances executed with the LP relaxation. At last, the last column *LP solution is integer?* tells us if the obtained solution of LP relaxation allows non-integer variables or not. In all instances, the optimal value of both models is the same, which means that the integrality gap is 1, that is, the null integrality gap. We can not state that the model has a null integrality gap but that the model has a null integrality gap for a large set of instances. Additionally, in some cases, the optimal solution allows non-integer variables. So, this means that, although the LP relaxation gave us good approximation solutions of the IP model, the solution may turn out to be integer infeasible, and if we round it, the solution may not be the optimal integer solution.

Table 6. Computational experiments 1 - instances' data

instance	planning horizon	number of workers	number of teams	workforce requirements	preferences' penalizations
1	31 days	30	1	req0	pref0
2	31 days	30	1	req1	pref0
3	31 days	30	1	req2	pref0
4	31 days	30	1	req3	pref0
5	31 days	30	1	req4	pref0
6	31 days	30	1	req5	pref0
7	31 days	30	1	req6	pref0
8	31 days	30	1	req7	pref0
9	31 days	30	1	req8	pref0
10	31 days	30	1	req9	pref0
11	31 days	30	1	req10	pref0
12	31 days	50	1	req0	pref1
13	31 days	50	1	req1	pref1
14	31 days	50	1	req2	pref1
15	31 days	50	1	req3	pref1
16	31 days	50	1	req4	pref1
17	31 days	50	1	req5	pref1
18	31 days	50	1	req6	pref1
19	31 days	50	1	req7	pref1
20	31 days	50	1	req8	pref1
21	31 days	50	1	req9	pref1
22	31 days	50	1	req10	pref1
23	31 days	75	1	req0	pref2
24	31 days	75	1	req1	pref2
25	31 days	75	1	req2	pref2
26	31 days	75	1	req3	pref2
27	31 days	75	1	req4	pref2
28	31 days	75	1	req5	pref2
29	31 days	75	1	req6	pref2
30	31 days	75	1	req7	pref2
31	31 days	75	1	req8	pref2
32	31 days	75	1	req9	pref2
33	31 days	75	1	req10	pref2
34	31 days	100	1	req0	pref3
35	31 days	100	1	req1	pref3
36	31 days	100	1	req2	pref3
37	31 days	100	1	req3	pref3
38	31 days	100	1	req4	pref3
39	31 days	100	1	req5	pref3
40	31 days	100	1	req6	pref3
41	31 days	100	1	req7	pref3
42	31 days	100	1	req8	pref3
43	31 days	100	1	req9	pref3
44	31 days	100	1	req10	pref3
45	31 days	150	1	req0	pref4
46	31 days	150	1	req1	pref4
47	31 days	150	1	req2	pref4
48	31 days	150	1	req3	pref4
49	31 days	150	1	req4	pref4
50	31 days	150	1	req5	pref4
51	31 days	150	1	req6	pref4
52	31 days	150	1	req7	pref4
53	31 days	150	1	req8	pref4
54	31 days	150	1	req9	pref4
55	31 days	150	1	req10	pref4
56	28 days	71	2	$req11_{team1}$	pref5
				$req11_{team2}$	
57	31 days	71	2	$req12_{team1}$	pref5
				$req12_{team2}$	
58	28 days	55	5	$req13_{team1}$	pref6
				$req13_{team2}$	
				$req13_{team3}$	
				$req13_{team4}$	
				$req13_{team5}$	

Table 7. Computational experiments 1 - results

instance	OPT_{int}	CPU$_{int}$ time	OPT_{frac}	CPU$_{frac}$ time	LP solution is integer?
1	329	00:26:93	329	00:20:20	yes
2	329	00:24:73	329	00:19:85	yes
3	322	00:24:49	322	00:20:07	yes
4	322	00:26:89	322	00:19:94	yes
5	366	00:27:25	366	00:19:98	yes
6	366	00:26:85	366	00:19:96	yes
7	364	00:26:03	364	00:19:65	yes
8	366	00:25:02	366	00:20:23	yes
9	343	00:24:27	343	00:19:98	yes
10	343	00:24:05	343	00:20:00	yes
11	345	00:24:24	345	00:19:72	yes
12	293	00:37:94	293	00:29:80	yes
13	293	00:36:71	293	00:29:91	yes
14	296	00:37:35	296	00:30:32	yes
15	296	00:37:19	296	00:29:93	yes
16	295	00:36:98	295	00:30:86	yes
17	295	00:37:15	295	00:30:08	yes
18	294	00:39:91	294	00:30:50	yes
19	295	00:37:92	295	00:30:36	yes
20	292	00:37:56	292	00:30:52	yes
21	292	00:36:93	292	00:29:69	yes
22	294	00:36:84	294	00:30:21	yes
23	293	00:55:72	293	00:43:96	yes
24	293	00:52:93	293	00:43:37	yes
25	296	00:54:84	296	00:43:01	yes
26	296	00:54:95	296	00:43:60	yes
27	295	00:52:90	295	00:43:57	yes
28	295	00:56:06	295	00:43:16	yes
29	294	00:51:29	294	00:43:53	yes
30	295	00:50:51	295	00:43:50	yes
31	292	00:52:37	292	00:46:89	yes
32	292	00:53:94	292	00:42:89	yes
33	294	00:52:32	294	00:43:26	yes
34	293	01:13:05	293	00:57:15	yes
35	293	01:14:72	293	00:57:05	yes
36	296	01:07:53	296	00:57:31	yes
37	296	01:07:97	296	00:56:66	yes
38	295	01:44:39	295	00:56:94	yes
39	295	01:57:52	295	00:56:92	yes
40	294	01:55:99	294	00:57:16	yes
41	295	01:58:96	295	00:57:46	yes
42	292	01:56:45	292	00:57:71	yes
43	292	01:57:10	292	00:57:74	yes
44	294	01:43:72	294	00:57:75	yes
45	293	02:19:79	293	01:27:43	yes
46	293	01:51:79	293	01:26:88	yes
47	296	02:00:22	296	01:28:97	yes
48	296	01:56:59	296	01:26:70	yes
49	295	02:04:18	295	01:26:19	yes
50	295	02:03:17	295	01:27:55	yes
51	294	02:02:82	294	01:26:52	yes
52	295	01:39:51	295	01:26:30	yes
53	292	02:01:27	292	01:27:11	yes
54	292	02:00:58	292	01:31:75	yes
55	294	02:00:58	294	01:31:95	yes
56	1107	00:53:23	1107	00:45:66	yes
57	1077	1:21:57	1077	00:59:21	no
58	1099	00:46:48	1099	00:38:67	yes

To evaluate the performance of the model with large instances, 15 randomly generated instances were tested, considering the planning horizon, the number of workers, and the preferences' penalizations equal in all cases, varying only the workforce's requirements. So, all instances have a planning horizon of a 31-day month and 500 workers organized in a single team. The preferences' penalizations follow the same logic mentioned above but are applied to 500 workers. Regarding workforce requirements, they were randomly generated between 0 and 50 or between 0 and 100, as can be seen in Table 8.

Table 8. Computational experiments 2 - instances' data

instance	planning horizon	number of workers	number of teams	workforce requirements
59	31 days	500	1	`randbetween(0,50)`
60	31 days	500	1	`randbetween(0,50)`
61	31 days	500	1	`randbetween(0,50)`
62	31 days	500	1	`randbetween(0,50)`
63	31 days	500	1	`randbetween(0,50)`
64	31 days	500	1	`randbetween(0,50)`
65	31 days	500	1	`randbetween(0,50)`
66	31 days	500	1	`randbetween(0,50)`
67	31 days	500	1	`randbetween(0,50)`
68	31 days	500	1	`randbetween(0,50)`
69	31 days	500	1	`randbetween(0,100)`
70	31 days	500	1	`randbetween(0,100)`
71	31 days	500	1	`randbetween(0,100)`
72	31 days	500	1	`randbetween(0,100)`
73	31 days	500	1	`randbetween(0,100)`

The combination of the memory RAM of the computer used in the computational experiments and the version of the IBM ILOG CPLEX Optimization Studio initially used, *i.e.* 12.7.0.0, makes it impossible to achieve any results when testing instances of Table 8, forcing a software version update. So, to test these 15 instances, the code is implemented in IBM ILOG CPLEX Optimization Studio 12.10.0.0.

The results are given in Table 9. Column *OPT* gives the optimal value of the IP model, followed by column *CPU time* that provides the process with time (minutes) for each one of the instances executed. In instance 65, the highest CPU time is achieved, having 744000 variables and 743988 constraints. So, considering the tested instances' size, the model presents a good performance, achieving good computational results.

Table 9. Computational experiments 2 - results

instance	OPT	CPU time
59	3414	06:47:00
60	3374	06:49:45
61	3438	06:32:41
62	3405	06:55:49
63	3360	07:01:94
64	3374	06:53:62
65	3387	07:12:48
66	3465	07:12:06
67	3348	06:56:46
68	3382	06:36:81
69	6772	06:51:57
70	6757	06:34:06
71	6770	06:26:12
72	6868	07:03:08
73	6768	06:31:35

Considering the solution given when executing real-data instance 56 and using Excel's tools, Figs. 4 and 5 illustrate the final schedule of team 1 and team 2, respectively. The CPU time in this scenario is lower than 1 min, having 190848 variables and 97615 constraints. The goal of reducing the processing time was obviously achieved.

When compared to schedules represented in Figs. 2 and 3, schedules of Figs. 4 and 5 appear to have less stability than the former. However, the penalization systems used to obtain them are different: the one used in the proposed method is more relaxed. So, consider, for example, a worker that prefers to work in the 07 h 00 shift. In the IP developed model, assigning this worker to the 07 h 00 shift or assigning the same worker to the 08 h 00 shift is identical because the penalization associated with these two time periods is the same.

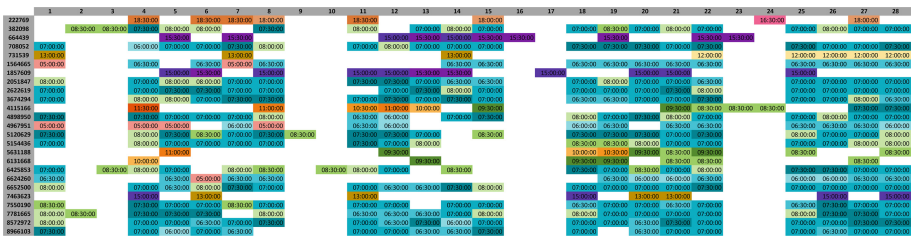

Fig. 4. Schedule of team 1 - IP solution

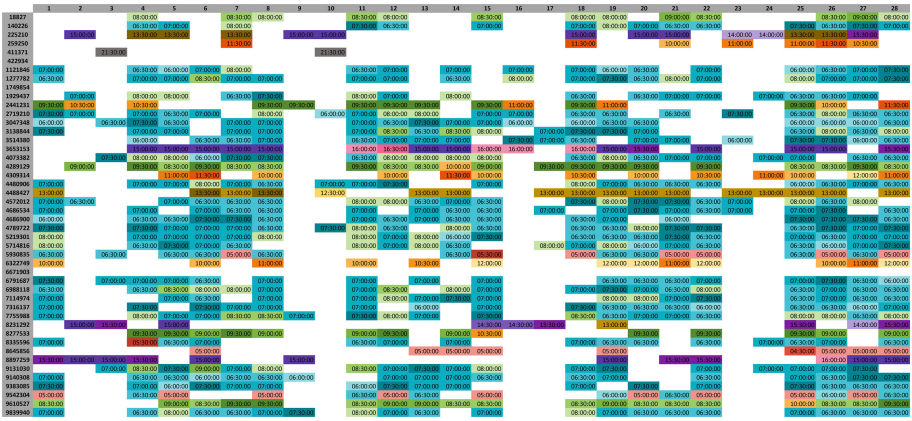

Fig. 5. Schedule of team 2 - IP solution

5 Conclusions

The personnel scheduling problem is a critical problem in the call center industry since call centers ensure communication between companies and customers. So, it is important for call centers to be staffed so that the waiting times experienced by customers are acceptable. In general, good and optimized schedules are advantageous for companies because they can avoid overstaffing, minimizing the total labor costs. To automate a manual scheduling process done in companies and, therefore, reduce the processing time, the main goal of this project was to contribute with a new solution approach that relies on exact methods. In this sense, an integer programming model for assigning work shifts to each employee in a 24/7 call center while considering their time preferences is proposed and implemented on IBM ILOG CPLEX Optimization Studio 12.7.0.0.

The model and its linear relaxation were tested to analyze the quality of the original model with both randomly generated and real-world instances. The integer programming model was proven to have a null integrality gap in all instances tested, which means that its formulation is strong. So, the proposed model is a good quality model. To evaluate the performance of the model when running large instances, more instances were tested, using a more updated version (version 12.10.0.0) of the chosen software since the older one was not able to run such large instances with the proposed method. The model was proven to have a good performance, achieving low CPU times considering the size of the tested instances.

An advantage of the developed model is that it can be applied to scheduling problems in other areas. It would be interesting to add more constraints in the proposed IP model for future work, such as scheduling of days-off and breaks and the possibility of workers changing shifts between them. It would also be interesting to implement the suggested solution method in an organization to

analyze some performance measures, like attendance and reactivity percentage. These topics can be seen as limitations of the proposed approach.

Funding. This work has been supported by FCT - Fundação para a Ciência e a Tecnologia within the R&D Units Project Scope UIDB/00319/2020.

References

1. Aksin, Z., Armony, M., Mehrotra, V.: The modern call center: a multi-disciplinary perspective on operations management research. Prod. Oper. Manag. **16**(6), 665–688 (2007)
2. Alfares, H.: Operator staffing and scheduling for an it-help call centre. Eur. J. Ind. Eng. **1**, 414–430 (2007)
3. Baker, K.R.: Workforce allocation in cyclical scheduling problems: a survey. J. Oper. Res. Soc. **27**(1), 155–167 (1976)
4. Bard, J.F., Binici, C., deSilva, A.H.: Staff scheduling at the united states postal service. Comput. Oper. Res. **30**(5), 745–771 (2003)
5. Bard, J.F., Purnomo, H.W.: Preference scheduling for nurses using column generation. Eur. J. Oper. Res. **164**(2), 510–534 (2005)
6. Bechtold, S.E., Brusco, M.J., Showalter, M.J.: A comparative evaluation of labor tour scheduling methods. Decis. Sci. **22**(4), 683–699 (1991)
7. den Bergh, J.V., Beliën, J., Bruecker, P.D., Demeulemeester, E., Boeck, L.D.: Personnel scheduling: a literature review. Eur. J. Oper. Res. **226**(3), 367–385 (2013)
8. Bhulai, S., Koole, G., Pot, A.: Simple methods for shift scheduling in multiskill call centers. Manuf. Serv. Oper. Manag. **10**, 411–420 (2008)
9. Cezik, M.T., L'Ecuyer, P.: Staffing multiskill call centers via linear programming and simulation. Manag. Sci. **54**(2), 310–323 (2008)
10. Cordone, R., Piselli, A., Ravizza, P., Righini, G.: Optimization of multi-skill call centers contracts and work-shifts. Serv. Sci. **3**, 67–81 (2011)
11. Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: a review of applications, methods and models. Eur. J. Oper. Res. **153**(1), 3–27 (2004). Timetabling and Rostering
12. Ertogral, K., Bamuqabel, B.: Developing staff schedules for a bilingual telecommunication call center with flexible workers. Comput. Ind. Eng. **54**(1), 118–127 (2008)
13. Felici, G., Gentile, C.: A polyhedral approach for the staff rostering problem. Manag. Sci. **50**(3), 381–393 (2004)
14. Maenhout, B., Vanhoucke, M.: A perturbation matheuristic for the integrated personnel shift and task re-scheduling problem. Eur. J. Oper. Res. **269**(3), 806–823 (2018)
15. Mohan, S.: Scheduling part-time personnel with availability restrictions and preferences to maximize employee satisfaction. Math. Comput. Model. **48**(11), 1806–1813 (2008)
16. Robbins, T.R., Harrison, T.P.: A stochastic programming model for scheduling call centers with global service level agreements. Eur. J. Oper. Res. **207**(3), 1608–1619 (2010)
17. Smet, P., Wauters, T., Mihaylov, M., Berghe, G.V.: The shift minimisation personnel task scheduling problem: a new hybrid approach and computational insights. Omega **46**, 64–73 (2014)
18. Thompson, G.M.: Labor scheduling: a commentary. Cornell Hotel Restaur. Adm. Q. **44**(5), 149–155 (2003)