*Research Article*

# An Integrated Approach for Shift Scheduling and Rostering Problems with Break Times for Inbound Call Centers

**Turgay Türker** [ID] [1] **and Ayhan Demiriz** [ID] [1,2]

[1]*Department of Industrial Engineering, Sakarya University, 54187 Sakarya, Turkey*
[2]*Department of Industrial Engineering, Gebze Technical University, Kocaeli 41400, Turkey*

Correspondence should be addressed to Ayhan Demiriz; ademiriz@gmail.com

It may be very difficult to achieve the optimal shift schedule in call centers which have highly uncertain and peaked demand during short time periods. Overlapping shift systems are usually designed for such cases. This paper studies shift scheduling and rostering problems for inbound call centers where overlapping shift systems are used. An integer programming model that determines which shifts to be opened and how many operators to be assigned to these shifts is proposed for the shift scheduling problem. For the rostering problem both integer programming and constraint programming models are developed to determine assignments of operators to all shifts, weekly days-off, and meal and relief break times of the operators. The proposed models are tested on real data supplied by an outsource call center and optimal results are found in an acceptable computation time. An improvement of 15% in the objective function compared to the current situation is observed with the proposed model for the shift scheduling problem. The computational performances of the proposed integer and constraint programming models for the rostering problem are compared using real data observed at a call center and simulated test instances. In addition, benchmark instances are used to compare our Constraint Programming (CP) approach with the existing models. The results of the comprehensive computational study indicate that the constraint programming model runs more efficiently than the integer programming model for the rostering problem. The originality of this research can be attributed to two contributions: (a) a model for shift scheduling problem and two models for rostering problem are presented in detail and compared using real data and (b) the rostering problem is considered as a task-resource allocation and considerably shorter computation times are obtained by modeling this new problem via CP.

## 1. Introduction

Service industry is known for its labor-intensive structure. Therefore, any improvement on labor costs will result in considerable savings in total cost. Moreover, effective labor schedules can lead to mutual satisfaction among the management, customers, and the staff in the service sector. In this regard, it is inevitable for such businesses to focus more on the effective use of the labor which is the most significant and the most expensive resource for them. Indeed, several studies on the problem of workforce scheduling also prove that reducing the employee cost expenses is beneficial to the firms [1].

Considering the direct communication with the customer side of the service sector, it is crucial to do the planning in the call centers where hundreds of operators work and where thousands of calls are processed. Since workforce cost is a significant component of the operating costs of the call centers, effective use of the workforce is vital to the call centers. In the work of Gans et al. [2], it is stated that labor costs are almost 70% of the total cost. According to Dietz [3], this ratio indicates that the economic efficiency of the operations is determined by the quality of the workforce planning process.

Several studies on the workforce planning of the call centers focus on generating workforce schedules meeting both the necessary service levels constraints and the requirements of some laws and regulations. While generating these schedules, it is extremely hard to find good solutions. There arise conflicting objectives when considering preferences of the employees, minimizing the costs, and meeting all the restrictions of the workplace [4]. As indicated in [5],
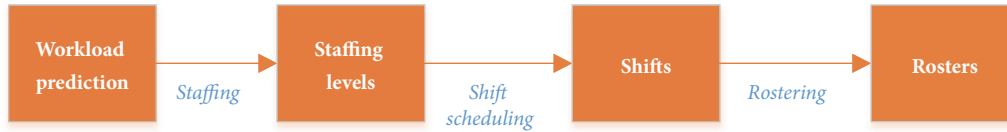
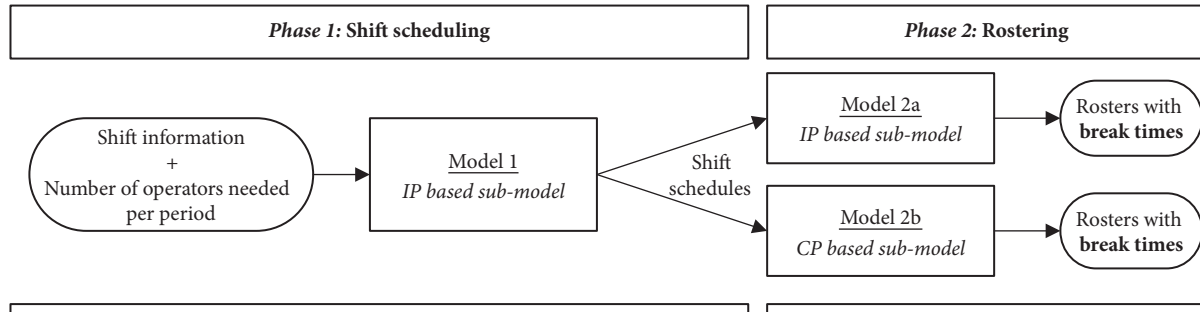FIGURE 1: Subproblems of workforce scheduling.



FIGURE 2: Subproblems considered within this study.

the workforce scheduling problem in a call center has four different subproblems which are shown in Figure 1.

The first subproblem is about forecasting the work load and number of calls received within certain periods (usually 30 minutes or 15 minutes [6]). The frequency and the number of calls received during the planning horizon are the main topic of this subproblem. The second subproblem, staffing, is about finding the required minimum number of operators depending on the incoming call rate. In other words, staffing is converting the estimated number of incoming calls to the operator requirements at periodic levels. The third subproblem focuses on choosing the optimal shifts that meet the demands for operators. With this subproblem, the shifts to be activated (opened) are also determined instead of simply assigning the operators to the available shifts. The fourth subproblem, rostering, deals with the assignment of the available operators to the shifts and determining the daily breaks and/or usually weekly days-off of the operators.

In shift scheduling, the organization of the shifts in the planning horizon is arranged. Schedules generally include nonoverlapping shifts. Allocation problem is easy to solve in this case. However, call centers may have overlapping shifts. Therefore, a shift scheduling system is crucial in call centers. In the rostering problem, an operator-shift matching is made based on definite constraints. Also, the days-off and the break times of each operator are determined. Here, we want to focus on the issue of determining the break times. As it is known, an operator usually works between seven and nine hours a day and (s)he needs breaks during this period. In their study, Sungur et al. [7] emphasized that operator-break assignments should be handled during the scheduling process. Besides, other studies in the literature state that only meal breaks are included in their problem formulations and that relief breaks should be taken into consideration as well [7].

Our aim in this study is to provide a workforce scheduling model including shift scheduling and rostering. The general

framework of the study is illustrated in Figure 2. First, we solve the shift scheduling problem by taking shift information and the staffing requirements as inputs into our integer programming (IP) model (Model 1). Staffing requirements means the minimum number of operators ensuring the service quality level within each period of the planning horizon. Afterwards, a rostering problem will be solved when the shifts are identified based on the solutions of the first problem.

Two different models are proposed for the rostering problem: traditional IP modeling approach (Model 2a) and constraint programming (CP) approach (Model 2b). With CP approach, the problem is handled as a task-source allocation and modeled with this perspective. The main contribution of this paper is to propose a CP model to solve the rostering problem in call center, as an alternative to the IP model.

The remainder of this paper is organized as follows. Section 2 introduces the related literature review on workforce scheduling problems. In Section 3, IP based model for shift scheduling problem and then IP and CP based models for rostering problem are introduced. Section 4 describes experimental study based on real data and test instances derived from the real data as well as the staff scheduling problem benchmark instances [8]. The paper is concluded, and the future directions are highlighted in Section 5.

## 2. Literature Review

There exist a lot of models developed in the literature for workforce scheduling problem [9]. Edie [10] analyzed the traffic delays in the tolls with an experimental study and formulized fixed queueing systems. Later, Dantzig [11] introduced the integer linear programming model with regard to Edie's [10] study. However, the workforce scheduling problem had gone through many evolutions in time.

The studies of Ernst et al. [4, 12] and the study of Brucker et al. [9] presented a detailed literature review on

the application areas of these type of problems. A summary of studies in call center workforce scheduling between 2003 and 2018 is presented in Tables 1(a) and 1(b) which depict some of the characteristics and modeling approaches of the call center workforce scheduling studies in the literature. The types of subproblems handled in these studies are also pointed in Tables 1(a) and 1(b). There is a lack of study in handling shift scheduling and rostering problems together, determining break times (especially relief breaks) and using the CP model for rostering in the literature.

### 2.1. Literature on Subproblems of Workforce Scheduling in Call Centers

#### 2.1.1. Call Forecasting.
The first subproblem of workforce scheduling problem is to forecast incoming calls within particular periods [5]. There are two different types of calls in the call centers: outbound and inbound calls. Planning the outbound calls is not difficult. Due to the uncertainty in the arrival rate and duration of the inbound calls, planning this type of calls is considered challenging [13].

Bianchi et al. compared Holt-Winters (HW) exponential weighted moving average smoothing model and Box-Jenkins (ARIMA) autoregressive integrated weighted moving model for call forecasting in [14]. Although a higher accuracy rate in HW for simple systems is achieved, it is observed that ARIMA gives more realistic results for more complicated systems. Taylor [15] compares the forecast accuracy of different time series including a version of Holt-Winters smoothing method and concludes that simple forecast techniques are not very successful in large processes.

#### 2.1.2. Staffing.
The arrival rate of the inbound calls in call centers varies in time. The number of operators should be planned carefully to be flexible enough to ensure the customer satisfaction. There are many studies in the literature on determining the optimal number of operators based on the arrival rate. The existing studies reveal that call center system can be regarded as a queueing system [16]. *M/M/S*, in other words Erlang C model, is the commonly used and the simplest call center queueing model due to obtaining closed form analytical solution [17]. The models developed to calculate the number of essential workforces for the call centers are usually alternative of Erlang C calculations [2, 16, 18].

Brown et al. present some statistical analyses of the operations of a call center using queueing models in [18]. They propose practical and beneficial mathematical models. The main inputs of mathematical models are the statistics such as the number of the operators, volume of the calls, service time, and holding time. The empirical analysis in [18] supports the preference of the Erlang models for forecasting the customer delays regardless of whether basic assumptions are satisfied or not for the Erlang models.

#### 2.1.3. Shift Scheduling.
When the work day requires more than one shift and some of them are overlapping, determining the number of workforce to be assigned to these shifts and planning the starting and ending times of these shifts become important problems. The main objective is to meet the workforce demand at each period with the minimum cost [19].

There are several studies conducted on different areas of the literature. In [20], Thompson proposed two different models for shift scheduling. While one of them proposes an approach based on the acceptable service level per period, the other one reveals an approach considering the average service level during the planning horizon. The developed models are MIP based. Atlason et al. [21] suggested that the model of Thompson [20] can be modeled by a simulation based approach. Although both works consider period-based monitoring and assignment like our approach, their performances are not studied in a multishift case found in call center settings.

Koole and Van der Sluis [19] mention in their study that the service level should be provided at the period level in the traditional approach and this is actually a hard constraint. Instead of this approach, they propose a model to meet the target of the general service level over the planning horizon. In [5], an easy model is implemented and it has a short computational time. They propose a method for multiskilled agent case which is different from our approach. In addition to these studies, Henderson and Berry [22] and Aykin [23, 24] provide some basic examples. The performances of these methods are not known in multishift settings like call centers. Indeed, they are not studied for the overlapping shifts.

#### 2.1.4. Rostering.
Well-constituted rosters have some advantages such as low cost, efficient use of the resources, increase in the employee satisfaction, and fair work load and allocation of the shifts [41]. Aykin [23] proposes an integer model for the rostering problems. His study includes multiple relief and meal break time windows, and different break variables are defined for each shift. The main problem has two different shifts. Each shift lasts for nine hours including the break times. The employees have a 30-minute lunch break and two 15-minute relief breaks. The break approach presented in his paper is crucial in terms of being a basis for the prospective studies. The break policy in our approach is inspired from this model.

Thompson [20] compares different shift and rostering models in terms of workforce cost, service level, and workforce utilization rate via simulation analyses. Our models consider only the cost minimization, but a multicriteria objective function is taken in [20].

Ertogral and Bamuqabel [34] introduce two different models for the shift scheduling problem by dividing the operators as flexible and nonflexible ones. However, they do not consider break times. Dietz [3] proposes a quadratic model for rostering problem and proves that this quadratic model yields better results compared to a classical IP model. Overlapping and multishift configurations are considered, but decisions for the break times are not included in [3].

Nah and Kim [46] propose a model of the rostering problem for a hospital reservation center. They found out that the developed model decreases the expected cost and the abandonment rate of the company by using an applied case study. Considering various types of costs, such as labor, waiting, and abandonment costs, increases the significance

TABLE 1

(a) Literature summary: some characteristics of workforce scheduling studies in call centers.

| Reviewed literature | Year | Considered subproblems | | | | Break(s) | |
|---|---|---|---|---|---|---|---|
| | | Call forecasting | Staffing | Shift scheduling | Rostering | M | R |
| Alfares [25] | 2007 | | √ | | √ | √ | |
| Ásgeirsson& Sigurðardóttir [26] | 2014 | | √ | | √ | | |
| Atlason et al. [27] | 2008 | | √ | √ | | | |
| Atlason et al. [21] | 2004 | | | √ | | | |
| Avradimis et al. [28] | 2008 | | √ | √ | | | |
| Avramidis et al. [29] | 2009 | | √ | | | | |
| Bhulai et al. [5] | 2008 | | √ | √ | | | |
| Canon [13] | 2007 | | √ | √ | √ | √ | |
| Castillo et al. [30] | 2009 | | √ | √ | | √ | √ |
| Cezik&L'Ecuyer [31] | 2008 | | √ | | | | |
| Defraeye et al. [32] | 2016 | | | √ | | | |
| Dietz [3] | 2011 | | √ | | √ | √ | |
| Dudin et al. [33] | 2013 | | √ | | | | |
| Ertogral&Bamuqabel [34] | 2008 | | √ | | √ | | |
| Excoffier [35] | 2016 | | | √ | | | |
| Gong et al. [17] | 2015 | | √ | | | | |
| Green et al. [36] | 2007 | | √ | | | | |
| Helber&Henken [37] | 2010 | | √ | √ | | | |
| Hojati [38] | 2010 | | | | √ | √ | |
| Ibrahim&L'Ecuyer [39] | 2013 | √ | | | | | |
| Ingolfsson et al. [40] | 2010 | | √ | | √ | √ | √ |
| Koole&van der Sluis [19] | 2003 | | √ | √ | | | |
| Kyngäs et al. [41] | 2012 | | | | √ | | |
| Liao et al. [42] | 2009 | | √ | | | | |
| Liu et al. [43] | 2018 | | | | √ | | |
| Mattia et al. [44] | 2017 | | √ | √ | | | |
| Millán-Ruiz&Hidalgo [45] | 2013 | √ | | | | | |
| Nah&Kim [46] | 2013 | | √ | | √ | √ | √ |
| Örmeci et al. [47] | 2014 | | | | √ | | |
| Pot et al. [48] | 2008 | | √ | | | | |
| Restrepo [49] | 2017 | | | | √ | √ | √ |
| Robbins&Harrison [50] | 2008 | | √ | √ | | | |
| Robbins&Harrison [51] | 2010 | | √ | | | | |
| Taskiran&Zhang [52] | 2017 | | | | √ | √ | |
| Wallace&Whitt [53] | 2005 | | √ | | | | |
| Weinberg et al. [54] | 2007 | √ | | | | | |

√: defined, otherwise undefined, M: meal break, R: relief break(s).

(b) Literature summary: modeling approaches of workforce scheduling studies in call centers.

| Reviewed literature | LP | IP | MIP | Queuing | Constructive heuristic | Other heuristic | Simulation | CP | Others |
|---|---|---|---|---|---|---|---|---|---|
| Alfares [25] | | √ | | √ | | | | | |
| Ásgeirsson& Sigurðardóttir [26] | | | √ | | | | | | |
| Atlason et al. [27] | | | √ | | | √ | √ | | |
| Atlason et al. [21] | | | √ | | | | √ | | |
| Avradimis et al. [28] | | | √ | | | | √ | | |
| Avramidis et al. [29] | | | √ | | | √ | √ | | |
| Bhulai et al. [5] | | | √ | | | √ | √ | | |
| Canon [13] | | | √ | | | √ | | √ | |
| Castillo et al. [30] | | | | √ | √ | | √ | | DEA |
| Cezik&L'Ecuyer [31] | √ | | | | | √ | √ | | |

(b) Continued.

| Reviewed literature | LP | IP | MIP | Queuing | Constructive heuristic | Other heuristic | Simulation | CP | Others |
|---|---|---|---|---|---|---|---|---|---|
| Defraeye et al. [32] | | | | | ✓ | | | | |
| Dietz [3] | | | | ✓ | | | | | |
| Dudin et al. [33] | | | | ✓ | | | | | |
| Ertogral&Bamuqabel [34] | | ✓ | | ✓ | | | ✓ | | AHP |
| Excoffier [35] | | | ✓ | | | | | | SP |
| Gong et al. [17] | | | | ✓ | | | | | |
| Green et al. [36] | | | | ✓ | | ✓ | ✓ | | |
| Helber&Henken [37] | | | ✓ | | | | ✓ | | |
| Hojati [38] | | | | | ✓ | | | | |
| Ibrahim&L'Ecuyer [39] | | | | | | | | | RM |
| Ingolfsson et al. [40] | | | ✓ | ✓ | | ✓ | | | |
| Koole&van der Sluis [19] | | | | ✓ | ✓ | | | | |
| Kyngäs et al. [41] | | | | | | ✓ | | | |
| Liao et al. [42] | | ✓ | | | | | ✓ | | SP |
| Liu et al. [43] | | ✓ | | | | ✓ | | | |
| Mattia et al. [44] | | ✓ | | | ✓ | | | | |
| Millán-Ruiz&Hidalgo [45] | | | | | | | | | NN |
| Nah&Kim [46] | | | ✓ | | | | | | |
| Örmeci et al. [47] | | | ✓ | | | | | | |
| Pot et al. [48] | | | ✓ | | | ✓ | | | |
| Restrepo [49] | | | | | | | | | SP |
| Robbins&Harrison [50] | | | | | ✓ | ✓ | ✓ | | |
| Robbins&Harrison [51] | | | ✓ | | | ✓ | ✓ | | SP |
| Taskiran&Zhang [52] | | ✓ | | | | | | | |
| Wallace&Whitt [53] | | | | ✓ | | | ✓ | | |
| Weinberg et al. [54] | | | | | | | | | BF |

✓: defined otherwise undefined, LP: linear programming, IP: integer programming, MIP: mixed integer programming, CP: constraint programming, DEA: data envelopment analysis, AHP: analytic hierarchy process, SP-stochastic programming, NN: neural network, RM: regression models, BF: bayesian forecasting.

of the work [46]. Break times are also determined. However, there are only two shifts for the planning horizon in [46]. Ásgeirsson and Sigurðardóttir [26] formulate a model for rostering problem and state that their model yields better results compared to a local search based algorithm. They implement their model for four different companies. High quality feasible staff schedules are achieved.

Örmeci et al. [47] examine the rostering problem in the call centers with the aim of balancing the operational costs, customer representative satisfaction, and customer service level objectives. It is also emphasized that call centers offer transportation services to its employees and since this constitutes a major part of the operational costs, they include the transportation requirement into the models. Especially, Canon's study [13] is very similar to our work from the perspective of usage of CP in workforce scheduling of call centers. The first three subproblems are studied in [13] by modeling MIP, CP, and Tabu Search. Unlike our study, Canon [13] studies just the staffing subproblem by modeling CP. However, only meal breaks are considered for the break scheduling and models are not tested on known benchmark sets in [13].

2.2. Constraint Programming. Constraint programming is a powerful tool for solving combinatorial search problems based on techniques such as artificial intelligence, operations research, and graph theory. The basic idea in the CP is that the user specifies constraints and solves these constraints through general-purpose solvers [55]. It is generally used to solve real-life problems in several application areas such as rostering, scheduling, and manufacturing [56].

Ernst et al. [4] present a review of workforce scheduling problems. In their study, workforce scheduling methods and techniques are reviewed. According to their study, classification of the different approaches contains five groups and CP is one of them. A review of the literature on workforce scheduling problems is presented in [1] too. CP is incorporated into solution techniques in their study. There are many studies with regard to CP in different application areas of workforce scheduling literature. Examples of the application areas are nurse scheduling/rostering [57–61], health care [62, 63], bus transportation [64], general [65–68], retail [69], military [70], and call center [13] (Canon's paper). We would like to draw attention to the lack of usage of CP in call center workforce scheduling studies.

In general, the main advantages of CP can be listed as follows: (1) it deals with heterogeneous constraints and nonconvex solution sets, (2) the domain size of the problem is independent, and (3) it enables the usage of optimization programming language (OPL) [71]. Furthermore, the CP models afford more useful analyses for real cases by requiring less computational effort [72]. Consequently, the usage of CP offers an ideal framework for various, complicated, and constrained workforce scheduling problems [1].

## 3. Models and Formulations

As it is seen in Tables 1(a) and 1(b), there are many different methods for workforce scheduling problems in the literature. The methods commonly used for call forecasting subproblem are ARIMA-HW [14, 15], regression models [39], queueing theory [16], simulation based algorithms [73], stochastic models [74, 75], Bayes approach [54, 76], and neural networks [45]. Researchers generally use mathematical (queueing) models [2, 18, 77] for staffing subproblem. Simulation based applications [21, 31] are also considered as an alternative in recent years. At first, MIP and IP models are considered for solving shift scheduling subproblem in many studies, but as the problem sizes, i.e., variable and constraint numbers, are increased and the problems become more complicated, it is observed that simulation based and heuristic/metaheuristic techniques are preferred [1]. Likewise, there are many studies which include IP, MIP, heuristics and hybrid models for rostering subproblem. Some of these studies are listed in Tables 1(a) and 1(b). Our proposed models are explained in this section.

*3.1. Shift Scheduling Model (Model 1).* The solution of shift scheduling problem mainly involves determining the number of the shifts to be chosen and the number of operators for these shifts. The result of the staffing problem indicates the assignment of the operators to fulfill the service level requirement. Minimizing the unused capacity means meeting the demand optimally. Besides, the decision of whether opening a shift or not will have its own associated costs, whereas unfulfillment of the demand might lead to the other costs. In this study, a shift scheduling model is proposed to minimize number of shifts, the excess and unfulfilled demands. The details of the model are as follows:

*Sets*

$D$: Set of days in the planning horizon ($\forall d \in D = \{1..|D|\}$))

$S$: Set of shifts in a day ($\forall s \in S = \{1..|S|\}$))

$P$: Set of periods in a work day ($\forall p \in P = \{1..|P|\}$))

*Parameters*

$C^P$: Working cost per period

$C^S$: Cost of opening a shift

$C^U$: Cost for unfulfilling the demand per period

$E_{sp}$: 1, if shift $s$ covers period $p$; 0, otherwise

$\gamma$: Maximum number of the operator to be assigned to a shift

$N_{dp}$: Number of operators needed at period $p$ of day $d$

*Decision Variables*

$a_{ds}$: Number of operators assigned to shift $s$ of day $d$ ($\forall d \in D, \forall s \in S$)

$p_{ds}$: Binary variable indicating if shift $s$ of day $d$ is chosen ($\forall d \in D, \forall s \in S$)

$u_{dp}$: Unfulfilled demand in period $p$ of day $d$ ($\forall d \in D, \forall p \in P$)

*Decision Expressions.* The number of the unnecessary operators in period $p$ of day $d$ is calculated as follows:

$$r_{dp} = \sum_{s \in S} \left( E_{sp}.a_{ds} \right) - N_{dp} \quad (\forall d \in D, \ \forall p \in P) \tag{1}$$

*Mathematical Model.* The mathematical model can be depicted as follows.

$$Minimize \quad \sum_{\substack{d \in D, \\ p \in P}} C^P.r_{dp} + \sum_{\substack{d \in D, \\ s \in S}} C^S.p_{ds} + \sum_{\substack{d \in D, \\ p \in P}} C^U.u_{dp} \tag{2}$$

$$subject\ to \quad \sum_{s \in S} \left( E_{sp} * a_{ds} \right) + u_{dp} \geq N_{dp} \tag{3}$$

$$(\forall d \in D, \ \forall p \in P)$$

$$p_{ds} \leq a_{ds} \leq \gamma p_{ds} \quad (\forall d \in D, \ \forall s \in S) \tag{4}$$

$$p_{ds}, E_{sp} \in \{0, 1\} \tag{5}$$

$$(\forall d \in D, \ \forall s \in S, \ \forall p \in P)$$

$$a_{ds}, u_{dp}, N_{dp}, \gamma \geq 0 \ and \ integer \tag{6}$$

$$(\forall d \in D, \ \forall s \in S, \ \forall p \in P)$$

The objective function of the proposed model, Eq. (2), is written based on the minimization of the operator costs, shift opening cost, and unfulfilled demand per shift, respectively. There are no differences among the operators in terms of productivity and skills (ability). The cost parameters are all the same for each operator in all of the proposed models. Equation (3) guarantees the assignment of the operators needed for each period of each day. Equation (4) enables the assignment of an operator if a shift is open, i.e., it makes sure that each shift has enough operators. Similarly, the maximum number of operators per shift will not be exceeded and "0" operator assignment to the opened shifts will be avoided with (4). Finally, (5) and (6) define the domain constraints.

*3.2. Rostering Models.* In this section, the details of the IP and CP models developed for the rostering problems are discussed.

*3.2.1. The IP Model (Model 2a).* We define the input data of an IP model for the rostering problem in this section.

*Sets*

$A$: Set of operators (agents) $(\forall a \in A = \{1..|A|\})$

$Y$: Set of break types $(\forall y \in Y = \{1..|Y|\})$

$B_{ys}$: Set of acceptable time intervals of break $y$ in shift $s$

$S_p^O$: Set of shifts that overlap with the time interval (period) $p$

*Parameters*

$C^H$: Cost of hourly work

$G_{ds}$: Total number of operators assigned to shift $s$ in day $d$

$H$: Number of work days in planning horizon.

$N_{dp}$: Number of operators needed at period $p$ of day $d$

$L_y$: Length of type $y$ break in periods (size)

$W_s$: Total working hours of shift $s$

*Decision Variables*

$x_{ads}$: Binary variable indicating if $a^{\text{th}}$ operator is assigned to shift $s$ in day $d$. $(\forall a \in A, \ \forall d \in D, \ \forall s \in S)$

$z_{adsyp}$: Binary variable indicating if $a^{\text{th}}$ operator is assigned to shift $s$ in day $d$ and the operator starts his/her break $y$ in period $p$. $(\forall a \in A, \ \forall d \in D, \ \forall s \in S, \ \forall y \in Y, \ \forall p \in P)$

*Decision Expression.* Total number of operators who have their break $y$ at period $p$ of day $d$ are computed as below.

$$W_{dyp} = \sum_{\overline{p}=p-L_y+1}^{p} \left( \sum_{\substack{a \in A, \\ s \in S_{\overline{p}}^O}} z_{adsy\overline{p}} \right) \tag{7}$$

$$(\forall d \in D, \ \forall y \in Y, \ \forall p \in P)$$

*Mathematical Model.* Mathematical model can be depicted as follows.

$$\text{minimize} \quad \sum_{\substack{a \in A, \\ d \in D, \\ s \in S}} \left( C^H . W_s . x_{ads} \right) \tag{8}$$

$$\text{subject to} \quad \sum_{s \in S} x_{ads} \leq 1 \quad (\forall a \in A, \ \forall d \in D) \tag{9}$$

$$\sum_{\substack{d \in D, \\ s \in S}} x_{ads} \leq (H-1) \quad (\forall a \in A) \tag{10}$$

$$x_{ads} - \sum_{p \in B_{ys}} z_{adsyp} = 0 \tag{11}$$

$$(\forall a \in A, \ \forall d \in D, \ \forall s \in S, \ \forall y \in Y)$$

$$\sum_a x_{ads} = G_{ds} \quad (\forall d \in D, \ \forall s \in S) \tag{12}$$

$$\sum_{s \in S_p^O} G_{ds} - \sum_y W_{dyp} \geq N_{dp} \tag{13}$$

$$(\forall d \in D, \ \forall p \in P)$$

$$x_{ads}, z_{adsyp} \in \{0, 1\} \tag{14}$$

$$(\forall a \in A, \ \forall d \in D, \ \forall s \in S, \ \forall y \in Y, \ \forall p \in P)$$

$$G_{ds}, W_{dyp}, N_{dp}, L_y, W_s \geq 0 \ and \ integer \tag{15}$$

$$(\forall d \in D, \ \forall s \in S, \ \forall y \in Y, \ \forall p \in P)$$

$$H > 1 \ and \ integer \tag{16}$$

The objective function, Eq. (8), is written based on the minimization of the total workforce cost. Equation (9) allows maximum one shift assignment to the operators each day over the planning horizon. Equation (10) enables taking a minimum of one day-off for the operators over the planning horizon. Equation (11) guarantees all the breaks of the operators and determines the frequency of these breaks. Equation (12) sets the number of the assigned operators equal to the number of operators in the shifts obtained as a result of the shift scheduling problem. That is, one of the solution values of the shift scheduling problem is used as a parameter of this rostering problem. Equation (13) ensures the sufficient number of operators working in each period of each day over the planning horizon. Finally, (14) through (16) defines the domain constraints.

*3.2.2. The CP Model (Model 2b).* The corner stones of any scheduling problem modeled by CP are the tasks [72]. We simplify the original scheduling problem by considering each break as a task, and assigning workforce resources to these tasks is the primary focus of our solution approach. By generating the solution of shift scheduling problem, the number of operators assigned each day is determined. The number of breaks of an operator assigned to any shift gives practically the number of tasks that will be performed by the operator. For instance, if operators have four breaks in a workday, the total number of the tasks will be four times the number of operators to be assigned during the planning period. Therefore, we are able to reduce the size of the scheduling problem significantly. In a way, we solve an equivalent scheduling problem.

| ID | Day | Type | Shift | Size | Start | End |
|----|-----|------|-------|------|-------|-----|
|    |     |      |       |      |       |     |

FIGURE 3: Data structure for tasks.

In the CP model, all the sets except the "acceptable time intervals of breaks in shifts" set which is explained in Model 2a will be utilized. In terms of parameters, while the cost per hour ($C^H$) per worker is considered in Model 2a, the cost per period ($C^P$) per worker is considered in Model 2b. The utilization of interval length for the total cost calculation is the reason of this difference. For instance, if 15-minute period system is taken into consideration, as there are four different periods in a one-hour working period, the periodical working cost is calculated as $C^P = C^S/4$. The sets and the parameters of the proposed model are as follows.

*Sets*

> $J$: Set of all tasks ($\forall j \in J = \{1..\tau|Y|\}$)
>
> $M$: Set of task dimension ($\forall m \in M = \{1..|M|\}$)
>
> $R$: Set of shifts ($\forall r \in R = \{1..|R|\}$)
>
> $O$: Task-resource allocation tuple
>
> $$O = \{\langle j, a \rangle \mid \forall j \in J, \forall a \in A\} \tag{17}$$
>
> $S^A$: Set of active shifts
>
> $$\left( S^A = \{\langle r, d, s \rangle \mid \forall r \in R, \forall d \in D, \forall s \in S\} \right) \tag{18}$$
>
> $S^P$: The start and end time of shifts
>
> $$\left( S^P = \{\langle s, i, e \rangle \mid \forall s \in S, \forall i \ \& \ e \in P\} \right) \tag{19}$$
>
> $T$: Tuple of tasks ($T = \{\langle j, d, y, s, m, i, e \rangle \mid \forall j \in J, \forall d \in D, \forall y \in Y, \forall s \in S, \forall m \in M, \forall i \ \& \ e \in P\}$)

*Parameters*

> $C^P$: Labor cost per period
>
> $G_{ds}$: Total number of operators assigned to shift $s$ in day $d$
>
> $\tau$: Total number of assignments in planning horizon
>
> $\delta_{ys}$: Start time of break type $y$ in shift $s$
>
> $\varepsilon_{ys}$: End time of break type $y$ in shift $s$

All possible tasks should be generated in the model before running the CP model. As it is seen in Figure 3, each task is identified with 7-field tuple layout ($T$). $j$ is used for the task index number. It expresses that type $y$ task is done in day $d$, shift $s$, and between period $i$ and period $e$ with an interval size of $m$. Here, the periods $i$ and $e$ which express the starting and ending periods of the related task type are derived from parameters $\delta_{ys}$ and $\varepsilon_{ys}$.

The starting and ending periods ($S^P$) and active shifts ($S^A$) are defined via 3-field tuple layout. While the elements of $S^P$

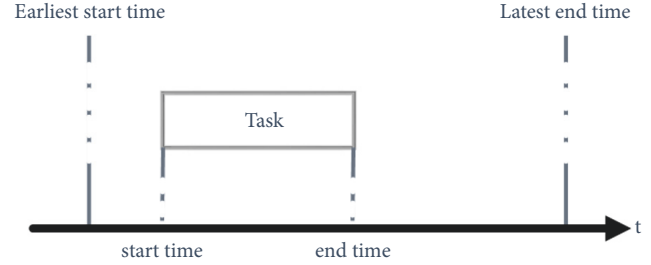

FIGURE 4: An interval variable.

tuple indicate that the shift $s$ starts in the $i^{th}$ period and ends in the $e^{th}$ period, elements of the $S^A$ tuple express that there is an assignment to $d^{th}$ day $s^{th}$ shift with $r$ index number. The $O$ tuple generated for task-source sharing means that resource $a$ is used for the task with $j$ index number.

*Decision Variables.* The time interval during which a task is completed in the CP can be represented with an `interval` decision variable. In Figure 4, an interval decision variable includes unique characteristics such as the earliest start time, the latest end time, and duration.

The position of a task between the earliest start time and the latest end time is not known at the beginning. The positions of intervals are determined by the sequence variables. Besides, the intervals can be considered optionally by using different constraints. The definitions of decision variables in our study are given as follows.

> $x_{ka}$: Decision variable for $a^{th}$ operator in shift $k$, interval variable which is optional between time periods $i$ and $e$ and with a size of $(e-i)$; $\forall k \in S^A, \forall a \in A$, $\forall i \ \& \ e \in S^P$
>
> $w_t$: Decision variable for task id $t$, interval variable which is optional between time periods $i$ and $e$ and with a size of $m$; $\forall t \in T$, ($i$, $e$, and $m$ are given as element of $T$)
>
> $z_o$: Decision variable for resource allocation of task $o$, interval variable which is optional; $\forall o \in O$
>
> $q_{ka}$: Decision variable represents a total order over a set of $z_o$, sequence variable; $\forall k \in S^A, \forall a \in A$.

*Cumulative Functions.* Ensuring that the number of active operators is equal to or less than the number of operators needed for each period is a complex constraint considering that there are overlapping shifts in our problem. With the proposed model, this functionality can be achieved quite easily via cumulative functions. In other words, the tasks and sources (operators) of these tasks are represented as a function of time. A cumulative function expression is represented by the OPL keyword `cumulFunction` and as
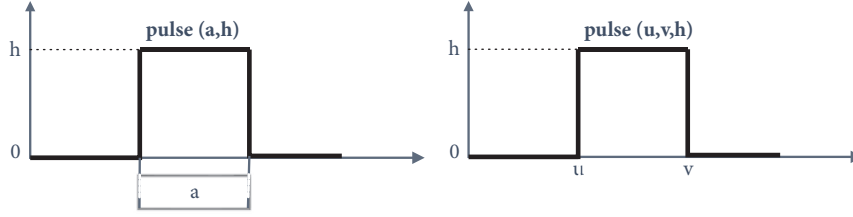
FIGURE 5: Example of pulse $(a, h)$ and pulse $(u, v, h)$.
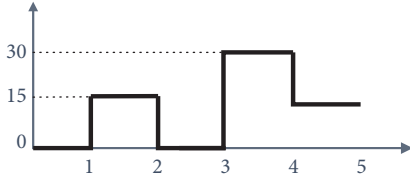


FIGURE 6: Example of cumulative function (stepwise $\{0 \longrightarrow 1, 1 \longrightarrow 15, 0 \longrightarrow 3, 30 \longrightarrow 4, 15 \longrightarrow 5\}$).

it is shown in Figure 5 when $a$ is expressed with interval decision variable, it is shown as $f = pulse(a, h)$ and when it is expressed with any $h$ parameter, it is shown as $f = pulse(u, v, h)$. The value of function $f = pulse(a, h)$ changes to $h$ at the start of an interval variable, while the value of function $f = pulse(u, v, h)$ changes to $h$ between times $u$ and $v$.

In short, as it is shown in Figure 6, a cumulative function expresses a step function that has a decreasing or increasing tendency in related to the fixed or varying time intervals. The functions from (20) to (22) express the number of operators assigned, number of the operators on a break, and number of the operators needed per period, respectively.

$$f_d^W = pulse\left(x_{ka}, 1\right)$$
$$\forall d \in D, \ \forall k \in S^A, \ \forall a \in A, \ k.d = d \tag{20}$$

$$f_d^B = pulse\left(w_t, 1\right) \quad \forall d \in D, \ \forall t \in T, \ t.d = d \tag{21}$$

$$f_d^R = pulse\left(p - 1, p, N_{dp}\right) \quad \forall d \in D, \ \forall p \in P \tag{22}$$

*The Constraint Programming Model.* Due to the differences in the functional construction of available modeling languages on the market, CP models are more dependent on the CP packages compared to the mathematical programming models. IBM ILOG CPLEX Optimization Studio 12.6 syntax is used as the modeling language in this study. Table 2 illustrates the definitions of the syntax used in the OPL model (see User's Manual for IBM ILOG CPLEX).

The CP model for the rostering problem is formulated as follows.

*Objective Function*

$$minimize \quad \sum_{\substack{k \in S^A, \\ a \in A}} \left(C^P.lengthOf\left(x_{ka}\right)\right) \tag{23}$$

Equation (23) is written based on the minimization of work-force cost. As it is underlined in Table 2, with the expression of $lengthOf$ the period length of $a^{th}$ operator's task in the $k^{th}$ assigned shift is found.

*Constraints*

*Presence (Logical Constraints)*

The presence of an optional interval can be determined using the presenceOf constraint. For instance, given that "$a$" and "$b$" are two optional intervals, if "$a$" is present, "$b$" is present as well. In a way this looks like Boolean expressions.

$$\sum_{a \in A} presenceOf\left(x_{ka}\right) = G_{ds} \quad \forall k \in S^A, \ \forall d \in D, \ \forall s \in S \tag{24}$$

$$\sum_{k \in S^A} presenceOf\left(x_{ka}\right) \leq 1 \quad \forall a \in A, \ \forall d \in D, \ k.d = d \tag{25}$$

$$\sum_{k \in S^A} presenceOf\left(x_{ka}\right) \leq (H - 1) \quad \forall a \in A \tag{26}$$

$$\sum_{\substack{t \in T, \\ o \in O}} presenceOf\left(z_o\right) = \left(presenceOf\left(x_{ka}\right)\right) \tag{27}$$

$$\forall a \in A, \ \forall k \in S^A, \ \forall y \in Y, \ t.j = o.j, \ o.a = a, \ t.s = k.s, \ t.d = i.d, \ t.y = y$$

$$\sum_{o \in O} presenceOf\left(z_o\right) = presenceOf\left(w_t\right) \quad \forall t \in T, \ t.j = o.j \tag{28}$$

TABLE 2: Syntax definitions used in OPL expressions and constraints.

| Syntax | Definition |
| --- | --- |
| *lengthOf* | Integer expression to access the length of an interval. |
| *noOverlap* | Constraint used to prevent intervals in a sequence from overlapping. |
| *presenceOf* | Constraint used to enforce the presence of intervals. |
| *synchronize* | Constraint used to synchronize the start and end of intervals. |
| *stepwise* | Keyword for stepwise linear functions. |

TABLE 3: List of CP constraints used for benchmark instances.

| Constraint on [8] | Definition of Constraint | Our Constraint Structure |
| --- | --- | --- |
| 1 | An employee cannot be assigned more than one shift on a single day | *Cumulative function* |
| 2 | A minimum amount of rest is required after each shift | *Cumulative function* |
| 3 | The maximum numbers of shifts of each type that can be assigned to employees | *Cumulative function* |
| 4 | Minimum and maximum work time | *Presence (Logical constraints)* |
| 5 | Maximum consecutive shifts | *Count function* |
| 6 | Minimum consecutive shifts | *Count function* |
| 7 | Minimum consecutive days off | *Count function* |
| 8 | Maximum number of weekends | *Presence (Logical constraints)* |
| 9 | Days-off | *Presence (Logical constraints)* |
| 10 | Cover requirements | *Cumulative function* |

Equation (24) sets the number of operators to be assigned to the shift during the day equal to the number of operators per shift obtained from the result of the shift scheduling problem. Equation (25) enables the assignment of the operators to maximum one shift in any day in the planning horizon. Equation (26) allows the operators to have at least one day-off in the planning horizon. Equation (27) guarantees that the operators get only one task (break) in each shift they are assigned. Equation (28) enables that each task is done by only one operator.

*Synchronize Formation.* A synchronization constraint (keyword `synchronize`) between an interval decision variable $a$ and a set of interval decision variables $B$ makes all present intervals in the set $B$ start and end at the same times as interval $a$, if it is present [78]. With (29), the start and end times of the allocations ($z_o$) are determined.

$$synchronize\left(w_t, \{z_1, \ldots, z_o : o \in O\}\right)$$
$$\forall t \in T, \ t.j = o.j \tag{29}$$

*Overlap Prevention.* A noOverlap constraint (keyword `noOverlap`) can be used to schedule the tasks that use specific resources. With (30), the assignment of an operator to another task while (s)he is performing a task during the same period is prevented.

$$noOverlap\left(q_{ka}\right) \quad \forall a \in A, \ \forall k \in S^A \tag{30}$$

*Cumulative Functions.* Equation (31) guarantees that enough operators are to be assigned to each period of each day in the planning horizon.

$$f_d^W - f_d^B \geq f_d^R \quad \forall d \in D \tag{31}$$

*Domain Constraints.* Finally, (32) defines the domain constraints.

$$G_{ds}, \delta_{ys}, \varepsilon_{ys}, \tau \geq 0 \ and \ integer$$
$$\forall d \in D, \ \forall s \in S, \ \forall y \in Y \tag{32}$$

In order to verify our CP model, we used benchmark test instances given in [8]. For this purpose, we rewrote the mathematical model in [8] with CP constructs. In other words, we developed a CP model derived from our work based on mathematical model proposed in [8]. We are only able to compare our CP approach with known results in the literature. The model in [8] is not given in detail in our paper due to space limitations. We prefer giving CP equivalents of the constraints in [8] in Table 3.

In addition to constraints given in Table 3, `synchronize` and `noOverlap` constraints are used to find solutions to the benchmark test instances. We do not use `count` constraint in our original CP model for the rostering problem. However, it is used for benchmark test instances (see Section 4.2.2). We report our findings in the following section.

## 4. Experimental Studies

### 4.1. A Real-Life Problem

*4.1.1. Problem Definition.* Real data obtained from an outsource call center supplying services to various companies are used in this study. The company provided us shift information and the minimum number of operators needed at each time interval to meet the call demand. The working day is divided into 15-minute periods. Therefore, there are 96 periods each day (i.e., $|P|=96$). Planning horizon is settled as seven days ($|D|=7$). Company provided data for a single

TABLE 4: Shift templates.

| Shift number (s) | Time range (24h) | Existing/ Additional shifts | Shift periods Start (i) | End (e) | Allotted periods for break type #1 | Allotted periods for break type #2 | Allotted periods for break type #3 | Allotted periods for break type #4 |
|---|---|---|---|---|---|---|---|---|
| 1 | 00-08 | ✓ | 1 | 32 | 3-8 | 11-17 | 21-26 | 27-32 |
| 2 | 08-16 | ✓ | 29 | 64 | 35-40 | 43-49 | 53-58 | 59-64 |
| 3 | 09-17 | ☑ | 33 | 68 | 39-44 | 47-53 | 57-62 | 63-68 |
| 4 | 10-18 | ✓ | 37 | 72 | 43-48 | 51-57 | 61-66 | 67-72 |
| 5 | 11-19 | ☑ | 41 | 76 | 47-52 | 55-61 | 65-70 | 71-76 |
| 6 | 12-20 | ✓ | 45 | 80 | 51-56 | 59-65 | 69-74 | 75-80 |
| 7 | 13-21 | ☑ | 49 | 84 | 55-60 | 63-69 | 73-78 | 79-84 |
| 8 | 14-22 | ☑ | 53 | 88 | 59-64 | 67-73 | 77-82 | 83-88 |
| 9 | 15-23 | ☑ | 57 | 92 | 63-68 | 71-77 | 81-86 | 87-92 |
| 10 | 16-00 | ✓ | 61 | 96 | 67-72 | 75-81 | 85-90 | 91-96 |

✓: existing shifts, ☑: additional shifts in the analyses, $S^P = \{\langle s, i, e \rangle \mid s \in S, i$ & $e \in P\}$.

planning horizon, i.e., seven days. The outline for the shifts is displayed in Table 4. There are five shifts ($s = 1, 2, 4, 6, 10$) during the weekdays that is from Monday ($d = 1$) to Friday ($d = 5$) and three shifts ($s = 1, 2, 10$) during the weekends ($d = 6, 7$) in this setup. We add five more possible shifts to this model and evaluate all of them regardless of weekday or weekend. With the ones added to the analyses, the possible shift set has ten members ($S$= {"00-08", "08-16", "09-17", "10-18", "11-19", "12-20", "13-21", "14-22", "15-23", "16-00"}) in our experimental setting.

Each operator has four break types during the shift (s)he is assigned ($|Y|$=4) in this firm. The break types of first, third, and fourth are relief break, the type of second break is meal break. The periods of all break types for each shift are shown in Table 4. There are one and half-hour break windows for the relief breaks and two- and a half-hour break windows for the meal break. All shifts are 8-hour long ($W_{\forall s \in S}$=8).

There are currently 76 operators working at this location ($|A|$=76). These operators can only work in one shift in a day and they have to take minimum one day-off within 7-day planning horizon. At most twenty operators can be assigned to a shift due to the capacity constraints in the shift scheduling problem ($\gamma = 20$). Relief and meal breaks should be taken at break time windows given in Table 4. Each relief break may last 15 minutes ($L_1, L_3$ and $L_4$=1), and meal break lasts for 30 minutes ($L_2$=2) for each operator. The times of break may differ for each operator in the call center.

The number of the workforce required for 15-minute periods for seven days ($N_{dp}$) in the planning horizon of the firm is also known. We illustrate the operator requirements for seven-day planning horizon in Figure 7.

First, the shift scheduling problem (Model 1) is solved for this organization. The solution of this problem will specify the number of operators to be assigned to the $s^{th}$ shift in the $d^{th}$ day ($G_{ds}$). However, these numbers will be adjusted pro rata in the rostering problem in order to meet the demand required per period during the breaks of the operators. The increase (adjustment) rate can be at most 30%. Besides, at most 26 operators can be assigned for a shift. Skipping this step leads to an infeasible solution for the rostering problem.

TABLE 5: Input data for Model 1 parameters.

| Parameters | Value | Unit |
|---|---|---|
| $C^H$ | 12 | TL/Operator per hour |
| $C^P$ | 3 | TL/Operator per period |
| $C^S$ | 300 | TL/Shift |
| $C^U$ | 100 | TL/Operator per period |
| $H$ | 7 | Days |

TL=Turkish liras

The updated $G_{ds}$, the tasks tuple ($T$), and active shifts tuple ($S^A$) will constitute the input for the rostering problem. The rest of the input data used is summarized in Table 5.

*4.1.2. Implementation and Computational Results.* The solutions of the shift scheduling problem are presented and then the results obtained from both models (IP/CP) of rostering problem are discussed in this section. All of the proposed models in this study are implemented by using IBM ILOG CPLEX Optimization Studio, which is available free of charge at IBM Academic Initiative web site for the academic users. We run our models on a laptop computer with Intel i5 Processor and 8 GB memory. The run time of CPLEX is set to 3600 seconds, i.e., one hour. Model 1 is solved both for the existing shift system and for the shifts included in the analyses. For both cases, an optimal solution is obtained, and the proposed case has shown an almost 15% improvement in the objective function compared to the existing case. The solution details of the models are shown in Table 6.

Day-shift assignment results for both cases are displayed in Table 7. The cells show the number of the required assignments for the related shift in the corresponding day. The cells including "0" indicate that the shift in the corresponding day will not be utilized and thus no assignments are required.

While 31 shifts are available for the existing case in the planning period, 27 shifts ($|R| = 27$) are utilized for the proposed case. Hence, four shifts are never used in planning in our case. Since the firm did not prefer unmet demand, all the demands in the corresponding planning horizon are met.
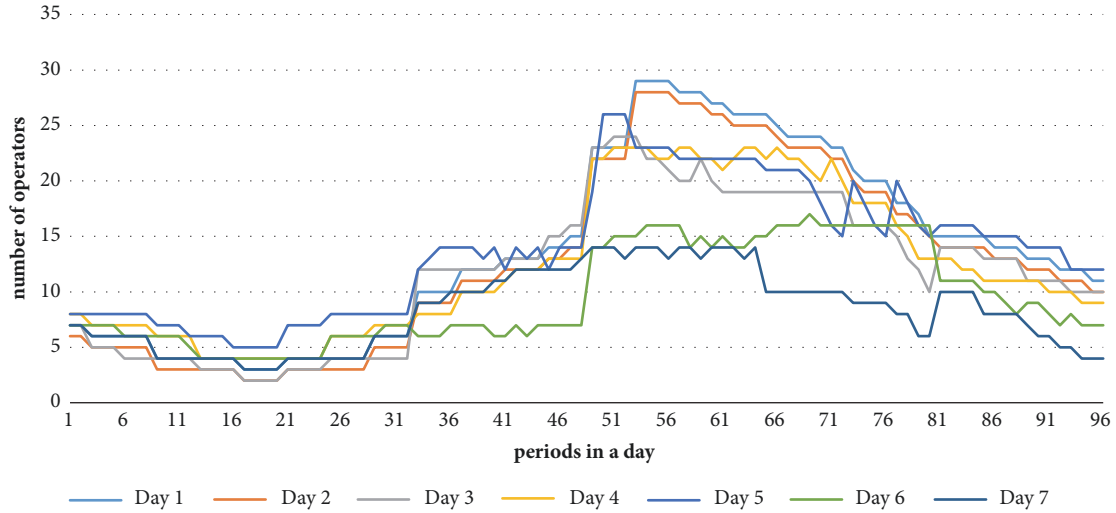
FIGURE 7: The number of operators needed at each period for 7-day planning horizon.

TABLE 6: Details of the Model 1.

| Problem | Objective | Time (s) | No. of constraints | No. of variables |
|---|---|---|---|---|
| Existing | 15,501 | 4.86 | 983 | 1,373 |
| Proposed | 13,437 | 4.77 | 952 | 1,373 |

TABLE 7: The numbers of the operators assigned to the shifts according to the days ($G_{ds}$).

| Days | Shifts | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | |
| | E | P | E | P | E | P | E | P | E | P | E | P | E | P | E | P | E | P | E | P |
| 1 | 7 | 7 | 13 | 15 | 0 | 0 | 9 | 0 | 0 | 0 | 7 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 15 |
| 2 | 6 | 6 | 14 | 17 | 0 | 0 | 8 | 0 | 0 | 0 | 6 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 14 |
| 3 | 7 | 7 | 15 | 13 | 0 | 0 | 7 | 0 | 0 | 11 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 15 |
| 4 | 8 | 8 | 13 | 13 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 13 |
| 5 | 8 | 8 | 14 | 14 | 0 | 0 | 6 | 0 | 0 | 0 | 7 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 16 |
| 6 | 7 | 7 | 16 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 11 |
| 7 | 7 | 7 | 14 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 10 |

E: Existing Case, P: Proposed Case

Even if there are unfulfilled demands, the cost of this would be bore and any necessary changes in the number of operators assigned to the shifts will be adjusted while passing to the rostering model.

The active shifts ($S^A$) to meet the demands are determined. The solutions for the operator assignment to active shifts, to their breaks, and to their days-off will be given. Phase 2 is only solved for the proposed case by determining the adjusted demand. There are not any changes in the program parameters for Model 2a, except adding a time constraint. In order to get more effective results in CP, some parameters can be adjusted. Most of the parameter settings are used in their default values. The parameters that are different from their default values are given in Table 8.

Since long computation time may be required for obtaining the optimized solution in CP optimizer, fail limit is set to 100,000. Number of workers is chosen as 1. If the number of

TABLE 8: Parameter setting for the CP solver.

| Parameter | Setting |
|---|---|
| Fail limit | 100,000 |
| Number of workers | 1 |
| Search phase | $x_{ka}$ |
| Search type | Restart |

workers is set to $n$, the CP optimizer engine creates $n$ workers, each in their own thread, and that will work in parallel to solve the problem. Using multiple workers requires more memory than using a single worker. A search phase allows us to specify the order of the search moves and the order in which the values must be tested. To prioritize the operator assignments to the shifts, we set the search phase as $x_{ka}$. "Restart" is chosen

TABLE 9: Model 2a and Model 2b details.

| Inc. rate (%) | $\tau$ | Model 2a (IP) | | | | Model 2b (CP) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Obj. | Time (s) | No. of const. | No. of variab. | Obj. | Time (s) (fail lim.) | No. of const. | No. of variab. |
| 15 | 365 | Infeasible[a] | | | | No solution[b] | | | |
| 20 | 376 | Infeasible[a] | | | | No solution[b] | | | |
| 21 | 382 | Infeasible[a] | | | | No solution[b] | | | |
| 22* | 386 | **37,056** | 877.23 | | | **37,056** | 7.07 (263.06) | 264,586 | 122,992 |
| 23* | 386 | **37,056** | 886.41 | | | **37,056** | 7.09 (263.65) | 264,586 | 122,992 |
| 24 | 390 | **37,440** | 898.57 | 677,214 | 876,709 | **37,440** | 6.16 (231.89) | 267,066 | 124,224 |
| 25 | 390 | **37,440** | 906.35 | | | **37,440** | 6.29 (229.30) | 267,066 | 124,224 |
| 30 | 409 | **39,264** | 915.33 | | | **39,264** | 6.75 (260.41) | 278,846 | 130,076 |
| 46 | 456 | **43,776** | 928.12 | | | **43,776** | 8.09 (209.65) | 307,986 | 144,552 |
| 47 | 462 | Infeasible[a] | | | | No solution[b] | | | |

The results in bold are proved to be optimal.
* The increase rate (Inc. rate) corresponding to the lowest optimal value of the case.
[a]No integer solution found.
[b]Search terminated by limit, no solution found.

as "search strategy" by considering the study of Régin [79]. The solution details of the models are illustrated in Table 9.

According to the results, the increase rate in the number of operators to be assigned to the $s^{\text{th}}$ shift in the $d^{\text{th}}$ day ($G_{ds}$) which arises as a result of shift scheduling stage will not be feasible for rostering stage if it is less than 22% or more than 46%. All the results in this range provide optimal results for both models, i.e., Models 2a and 2b of rostering problem. If the rate is 22% or 23%, there will be a total of $\tau$=386 assignments and the lowest optimal solution (386 x 12 TL/h x 8 h=37,056TL) will be reached. It is understood that the operators have approximately 1.92 day/week days-off.

Model 2b results in an impressively better performance than Model 2a in terms of solution time. In Table 9, while the values outside the parentheses in CP time column show the time to reach the optimum result, the values in parentheses show the time to reach the fail limit. While Model 2a reaches the minimum optimal result in 877.23 seconds, Model 2b reaches the same result in 7.07 seconds. When all cases are considered, it can be seen that the time to reach the optimal results in Model 2a is almost 130 times more than that in Model 2b. While the total number of constraints and variables does not depend on the increase rate in Model 2a, it changes in Model 2b, because the total number of tasks ($|J| = \tau|Y|$) increases in proportion to the total number of the assignments.

*4.2. Experimental Trials.* In this section, the proposed models for the rostering problem are tested on simulated instances and benchmark test instances in [8].

*4.2.1. Results from Simulated Test Instances.* We generated eleven test instances by varying the demand within certain limits per period from small to large to compare the performances of rostering models. First, we solved each test instance via Model 1 to figure out what shifts should be used. Table 10 presents the input and output values and test results

of Model 1. Then, by increasing the number of operators assigned to the shifts in certain rates, we generated the inputs for the rostering problem by balancing the total number of operators and the total number of assignments. Hence, the number of operators varied between 27 and 185 for the test instances.

Table 10 summarizes the results of the computations of Model 1. An optimal result is found for all instances, and objective function values, solution times, the numbers of the constraints, and the variables are shown in Table 10. The numbers of the constraints and the decision variables are the same for all instances. Average solution time is 4.43 seconds.

Table 11 compares the computational results of the two models for the rostering problem. The increase rate varies between 10% and 30% by 5% steps. The optimal solution values for all instances are shown in Table 11. For example, the optimal solution is reached with a 26% increase rate, and 141 operators are assigned to active shifts in the planning horizon. Results obtained between 26% and 30% increase rates are optimal and size of the operator set is equal to 27 for the first instance.

When both models are compared in terms of all instances, it can be inferred that Model 2b is much more efficient than Model 2a. The IP model contains a noticeably larger number of constraints and decision variables in all problems than the CP model. Moreover, the number of variables is more than the number of constraints in IP model. On the other hand, the number of the constraints is more than that of the variables in CP. While the numbers of the constraints and the variables are independent of the increase rate for each instance in IP model, the variable and the constraint sizes are in direct relation to the total number of assignments in CP model.

The proposed CP model can provide a feasible solution within seconds. When all cases are considered, it is seen that the time to reach the optimal results in IP is almost 290 times more than that of CP. In the biggest problem (the eleventh

TABLE 10: Information on test instances and details of Model 1.

| Ins. | Operator demand per period | | | $\gamma$ | Total number of the opened shifts | $\tau$ | Obj. | Time (s) | No. of const. | No. of variab. |
|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Ave. | Max | | | | | | | |
| 1 | 0 | 3.54 | 9 | 10 | 23 | 100 | **9,360** | 4.62 | 952 | 1,373 |
| 2 | 1 | 5.51 | 13 | 10 | 26 | 147 | **10,806** | 4.97 | 952 | 1,373 |
| 3 | 1 | 7.58 | 18 | 10 | 27 | 196 | **11,625** | 4.27 | 952 | 1,373 |
| 4 | 3 | 10.82 | 23 | 15 | 26 | 274 | **12,282** | 3.97 | 952 | 1,373 |
| 5 | 2 | 11.40 | 26 | 15 | 28 | 299 | **14,121** | 4.30 | 952 | 1,373 |
| 6 | 2 | 11.40 | 26 | 20 | 27 | 299 | **13,821** | 4.68 | 952 | 1,373 |
| 7 | 3 | 14.07 | 32 | 30 | 27 | 367 | **14,973** | 4.62 | 952 | 1,373 |
| 8 | 3 | 14.07 | 32 | 30 | 27 | 367 | **14,973** | 4.58 | 952 | 1,373 |
| 9 | 4 | 20.99 | 47 | 40 | 27 | 546 | **18,201** | 4.07 | 952 | 1,373 |
| 10 | 4 | 22.80 | 52 | 40 | 27 | 598 | **19,542** | 4.07 | 952 | 1,373 |
| 11 | 5 | 26.66 | 60 | 40 | 27 | 701 | **21,648** | 4.62 | 952 | 1,373 |

The results in **bold** are proved to be optimal.

instance), the IP model cannot provide a solution within the specified run time (3600 seconds), and CP model can give a solution only in 20 seconds. A large memory usage of 6.3 GB for this instance is seen in CP model. It cannot be found a solution with CP by using current limits and settings in larger models. However, we can say that the feasible solutions can be obtained with IP by increasing the run time and with CP by using a computer with a higher memory.

*4.2.2. Results of Benchmark Test Instances (from http://www .schedulingbenchmarks.org/).* In order to test our CP approach on publicly available benchmark test instances, we modified our model accordingly. Table 12 compares the computational results of models in [8] and our proposed CP model. The instances are available for download at http://www.cs.nott.ac.uk/~tec/NRP/, where all the required information on each instance, best solutions, and lower bounds is also available. Models studied in [8] are an Ejection Chain metaheuristic, a Branch and Price method, and an Integer Programming formulation coded on Gurobi 5.6.3. As seen in Table 12, the instances range from very small to very large.

As it is seen from Table 12, the branch and price method is very effective for small and medium size benchmark test instances. However, it fails to return solutions for larger instances, instance 13 and beyond, due to the lack of memory issues. On the other hand, our CP approach is able to return solutions for Instances 14-19. Our approach also fails due to the memory issues for instances between 20 and 24. For these instances Gurobi 5.6.3. is not able to find solutions either. Ejection Chain metaheuristics method yielded some solutions for the larger instances though not necessarily good solutions.

Our approach yielded the optimal solutions for the first four instances too within very reasonable run times compared to the others. For the larger instances, near optimal or near lower bound solutions are found. Nevertheless, it achieves better results than Ejection Chain metaheuristics for the instances that feasible solutions can be found. In

general, Gurobi 5.6.3 found the best near optimal solutions. However, our approach yielded the best results for instances 15, 18, and 19. Therefore, we can conclude that our approach is competitive enough in general with methods discussed in [8].

## 5. Conclusion

Call centers are business units where workforce is an expensive resource and an extensively used one. The main objective in these business units is to ensure the maximum customer and employee satisfaction via the minimum operational and workforce costs. In order to meet this objective, the firms have to utilize several techniques. Workforce scheduling models can be regarded as one of the most useful tools in this regard.

This paper addresses an integrated solution for the shift scheduling and the rostering problems within call centers. Shift scheduling problem is formulated as an IP model. We introduce an integer programming and a constraint programming model to solve the rostering problem. Assignments of operators to proposed shifts, weekly days-off, and meal and relief break times of the operators are determined with these rostering models. It is shown that even a detailed rostering problem can be easily regarded as a task-resource assignment problem with CP approach.

The proposed models are tested by using real-life data, test problems generated with a series of scenarios, and benchmark test instances. Cross-comparison through experimental results is conducted. While comparing these results, the focus is mostly on the models developed for rostering problems. The results indicate that CP modeling is more preferable than the IP modeling in terms of facilitation, flexibility, and high speed in computation time. Moreover, it is observed by analyzing benchmark results that CP can be utilized effectively for solving general workforce scheduling problems too. The advantage of CP over IP is related to the fact that the declarative language of logic has an explanatory power. The growth in the size of the models does not affect CP in terms of finding solutions. On the other hand, temporal

TABLE 11: Comparison of Model 2a and Model 2b for test problems.

| Ins. | Inc. rate (%) | $\tau$ | $|A|$ | Model 2a (IP) | | | | Model 2b (CP) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Obj. | Time (s) | No. of const. | No. of variab. | Obj. | Time (s) (fail lim.) | No. of const. | No. of variab. |
| 1 | 10 | 123 | 27 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 127 | 27 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 129 | 27 | Infeasible[a] | | | | No solution[b] | | | |
| | 25 | 134 | 27 | Infeasible[a] | | | | No solution[b] | | | |
| | 26* | 141 | 27 | **13,536** | 139.66 | 331,764 | 444,186 | **13,536** | 0.34 (65.31) | 39,846 | 17,034 |
| | 30 | 142 | 27 | **13,632** | 131.67 | 331,764 | 444,186 | **13,632** | 0.39 (72.13) | 40,074 | 17,146 |
| 2 | 10 | 173 | 38 | Infeasible[a] | | | | No solution[b] | | | |
| | 15* | 186 | 38 | **17,856** | 222.32 | 409,314 | 541,283 | **17,856** | 0.75 (128.84) | 70,969 | 30,992 |
| 3 | 10 | 223 | 51 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 240 | 51 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 243 | 51 | Infeasible[a] | | | | No solution[b] | | | |
| | 23* | 257 | 51 | **24,672** | 367.77 | 500,964 | 656,034 | **24,672** | 0.96 (194.44) | 124,906 | 56,210 |
| | 25* | 257 | 51 | **24,672** | 362.65 | 500,964 | 656,034 | **24,672** | 1.04 (198.79) | 124,906 | 56,210 |
| 4 | 10 | 316 | 66 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 330 | 66 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 342 | 66 | Infeasible[a] | | | | No solution[b] | | | |
| | 25 | 350 | 66 | Infeasible[a] | | | | No solution[b] | | | |
| | 26* | 357 | 66 | **34,272** | 566.91 | 606,714 | 788,439 | **34,272** | 1.20 (259.00) | 213,933 | 99,108 |
| | 30 | 372 | 66 | **35,712** | 559.43 | 606,714 | 788,439 | **35,712** | 1.26 (188.05) | 222,033 | 103,128 |
| 5 | 10 | 344 | 78 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 360 | 78 | Infeasible[a] | | | | No solution[b] | | | |
| | 19* | 371 | 78 | **35,616** | 778.32 | 691,314 | 894,363 | **35,616** | 2.65 (305.49) | 262,823 | 121,604 |
| | 20* | 371 | 78 | **35,616** | 834.24 | 691,314 | 894,363 | **35,616** | 2.49 (301.74) | 262,823 | 121,604 |
| 6 | 10 | 338 | 78 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 358 | 78 | Infeasible[a] | | | | No solution[b] | | | |
| | 19* | 367 | 78 | **35,232** | 856.23 | 691,314 | 894,363 | **35,232** | 2.67 (317.92) | 259,342 | 120,184 |
| | 20* | 367 | 78 | **35,232** | 861.55 | 691,314 | 894,363 | **35,232** | 2.55 (311.13) | 259,342 | 120,184 |
| 7 | 10 | 413 | 90 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 433 | 90 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 448 | 90 | Infeasible[a] | | | | No solution[b] | | | |
| | 22* | 459 | 90 | **44,064** | 1074.75 | 775,914 | 1,000,287 | **44,064** | 3.79 (313.65) | 365,902 | 171,936 |
| | 25 | 472 | 90 | **45,312** | 1133.63 | 775,914 | 1,000,287 | **45,312** | 4.05 (297.34) | 375,418 | 176,668 |
| 8 | 10 | 413 | 109 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 433 | 109 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 448 | 109 | Infeasible[a] | | | | No solution[b] | | | |
| | 22* | 459 | 109 | **44,064** | 1440.44 | 909,864 | 1,168,000 | **44,064** | 5.38 (416.09) | 441,978 | 207,846 |
| | 25 | 472 | 109 | **45,312** | 1616.64 | 909,864 | 1,168,000 | **45,312** | 4.59 (388.22) | 453,470 | 213,566 |
| 9 | 10 | 611 | 135 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 640 | 135 | Infeasible[a] | | | | No solution[b] | | | |
| | 20* | 667 | 135 | **64,032** | 2670.55 | 1,093,164 | 1,397,502 | **64,032** | 9.03 (504.03) | 773,218 | 370,138 |
| 10 | 10 | 666 | 160 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 703 | 160 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 724 | 160 | Infeasible[a] | | | | No solution[b] | | | |
| | 25 | 754 | 160 | Infeasible[a] | | | | No solution[b] | | | |
| | 26* | 768 | 160 | **73,728** | 2967.78 | 1,269,414 | 1,618,177 | **73,728** | 12.44 (832.95) | 1,045,410 | 503,232 |
| | 30 | 789 | 160 | **75,744** | 3228.78 | 1,269,414 | 1,618,177 | **75,744** | 12.89 (826.12) | 1,072,542 | 516,756 |

TABLE 11: Continued.

| Ins. | Inc. rate (%) | $\tau$ | $|A|$ | Model 2a (IP) | | | | Model 2b (CP) | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| | | | | Obj. | Time (s) | No. of const. | No. of variab. | Obj. | Time (s) (fail lim.) | No. of const. | No. of variab. |
| 11 | 10 | 780 | 185 | Infeasible[a] | | | | No solution[b] | | | |
| | 15 | 817 | 185 | Infeasible[a] | | | | No solution[b] | | | |
| | 20 | 852 | 185 | Timeout[c] | | | | No solution[b] | | | |
| | 25* | 889 | 185 | Timeout[c] | | | | **85,344** | 20.67 (930.82) | 1,387,842 | 671,406 |

The results in **bold** are proved to be optimal.
*The increase rate (Inc. rate) corresponding to the lowest optimal value of the case.
[a]No integer solution found.
[b]Search terminated by limit, no solution found.
[c]Timeout (3600 seconds).

TABLE 12: Comparison of our model and other models in [8] for benchmark instances.

| Ins. | W | St. | Sh. | LB | Ejection Chain | | Branch and Price | | Gurobi 5.6.3 | | Our Proposed CP Model | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | | | | | 10 min. | 60 min. | Sol. | Time (s) | Sol. | Time (s) | Sol. | Time (s) | First Time (s) |
| #1 | 2 | 8 | 1 | 607 | **607** | **607** | **607** | 0.27 | **607** | 1.62 | **607** | 0.75 | 0.75 |
| #2 | 2 | 14 | 2 | 828 | 923 | 837 | **828** | 0.13 | **828** | 5.22 | **828** | 62.69 | 62.69 |
| #3 | 2 | 20 | 3 | 1001 | 1003 | 1003 | **1001** | 0.45 | **1001** | 13.54 | **1001** | 158.33 | 158.33 |
| #4 | 4 | 10 | 2 | 1716 | 1719 | 1718 | **1716** | 1.50 | **1716** | 158.99 | **1716** | 518.48 | 518.48 |
| #5 | 4 | 16 | 2 | 1143 | 1439 | 1358 | 1160 | 25.61 | **1143** | 1520.24 | 1180 | 3600.15 | 722.62 |
| #6 | 4 | 18 | 3 | 1950 | 2344 | 2258 | 1952 | 10.46 | **1950** | 440.93 | 2014 | 3600.59 | 3588.83 |
| #7 | 4 | 20 | 3 | 1056 | 1284 | 1269 | 1058 | 93.73 | **1056** | 2152.48 | 1091 | 3600.50 | 3196.73 |
| #8 | 4 | 30 | 4 | 1281 | 2529 | 2260 | 1308 | 3600.00 | 1323 | 3599.83 | 1522 | 3601.76 | 3581.42 |
| #9 | 4 | 36 | 4 | 247 | 474 | 463 | 439 | 76.99 | 439 | 3599.85 | 452 | 3602.85 | 3164.02 |
| #10 | 4 | 40 | 5 | 4631 | 4999 | 4797 | **4631** | 113.44 | **4631** | 244.20 | 4666 | 3604.51 | 3572.05 |
| #11 | 4 | 50 | 6 | 3443 | 3967 | 3661 | **3443** | 19.11 | **3443** | 109.92 | 3455 | 3610.79 | 3507.97 |
| #12 | 4 | 60 | 10 | 4040 | 5611 | 5211 | 4046 | 1336.4 | **4040** | 2303.84 | 4409 | 3676.93 | 2998.03 |
| #13 | 4 | 120 | 18 | 1346 | 8707 | 3037 | Out of Memory | | 3109 | 3600.55 | | Out of Memory | |
| #14 | 6 | 32 | 4 | 1277 | 2542 | 1847 | Out of Memory | | 1280 | 3600.13 | 1345 | 3606.70 | 3531.89 |
| #15 | 6 | 45 | 6 | 3806 | 6049 | 5935 | Out of Memory | | 4964 | 3600.00 | 4548 | 3625.62 | 3317.59 |
| #16 | 8 | 20 | 3 | 3211 | 4343 | 4048 | 3323 | 265.02 | 3233 | 3599.99 | 3332 | 3601.62 | 3580.48 |
| #17 | 8 | 32 | 4 | 5726 | 7835 | 7835 | Out of Memory | | 5851 | 3600.00 | 5911 | 3607.98 | 3569.21 |
| #18 | 12 | 22 | 3 | 4351 | 6404 | 6404 | Out of Memory | | 4760 | 3599.99 | 4668 | 3604.73 | 3587.60 |
| #19 | 12 | 40 | 5 | 2945 | 6522 | 5531 | Out of Memory | | 5420 | 3605.90 | 4900 | 3626.12 | 3404.58 |
| #20 | 26 | 50 | 6 | 4743 | 23531 | 9750 | Out of Memory | | - | 3600.05 | | Out of Memory | |
| #21 | 26 | 100 | 8 | 20868 | 38294 | 36688 | Out of Memory | | - | 3600.21 | | Out of Memory | |
| #22 | 52 | 50 | 10 | - | - | 516686 | Out of Memory | | - | 3600.19 | | Out of Memory | |
| #23 | 52 | 100 | 16 | - | - | 54384 | Out of Memory | | - | 3600.43 | | Out of Memory | |
| #24 | 52 | 150 | 32 | ? | - | 156858 | Out of Memory | | Out of Memory | | | Out of Memory | |

Known optimal solutions are in **bold.**
W= Number of Weeks, St.= Number of Staff, Sh.= Number of Shift Type, LB=Best Known Lower Bound.

constraints prevent IP models for finding a feasible solution. One of the limitations of CP is exceeding the memory size due to the increase in size of model. In such occasions, CP models cannot provide any solutions because of the inadequacies in resources. Even though a modest laptop with 8 GB of RAM has been used in this study, some CP models are run on computers with larger memory like 48 GB in the literature. We expect that if our CP model is run on such a system, it will find feasible even optimal solutions within acceptable run time limits for larger benchmark instances.

Additional real-life constraints, e.g., an operator who works in a specified shift of a related day may not work in a prespecified shift in the following day or if the operator has a break at a certain period, the next break time should be after prespecified periods etc., can be added to the model in future studies. Multiskill rostering problem also is left to be studied in a follow-up work.

## Data Availability

The data used in our experiments are available upon request from the corresponding author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

[1] J. van den Bergh, J. Beliën, P. de Bruecker, E. Demeulemeester, and L. de Boeck, "Personnel scheduling: a literature review," *European Journal of Operational Research*, vol. 226, no. 3, pp. 367–385, 2013.

[2] N. Gans, G. Koole, and A. Mandelbaum, "Telephone call centers: tutorial, review, and research prospects," *Manufacturing & Service Operations Management*, vol. 5, no. 2, pp. 79–141, 2003.

[3] D. C. Dietz, "Practical scheduling for call center operations," *Omega* , vol. 39, no. 5, pp. 550–557, 2011.

[4] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: a review of applications, methods and models," *European Journal of Operational Research*, vol. 153, no. 1, pp. 3–27, 2004.

[5] S. Bhulai, G. Koole, and A. Pot, "Simple methods for shift scheduling in multiskill call centers," *Manufacturing & Service Operations Management*, vol. 10, no. 3, pp. 411–420, 2008.

[6] Z. Aksin, M. Armony, and V. Mehrotra, "The modern call center: a multi-disciplinary perspective on operations management research," *Production Engineering Research and Development*, vol. 16, no. 6, pp. 665–688, 2007.

[7] B. Sungur, C. Özgüven, and Y. Kariper, "Shift scheduling with break windows, ideal break periods, and ideal waiting times," *Flexible Services and Manufacturing Journal*, vol. 29, no. 2, pp. 203–222, 2017.

[8] T. Curtois and R. Qu, "Technical report: Computational results on new staff scheduling benchmark instances," Tech. Rep., 2017.

[9] P. Brucker, R. Qu, and E. Burke, "Personnel scheduling: models and complexity," *European Journal of Operational Research*, vol. 210, no. 3, pp. 467–473, 2011.

[10] L. C. Edie, "Traffic Delays at Toll Booths," *Journal of the Operations Research Society of America*, vol. 2, no. 2, pp. 107–138, 1954.

[11] G. B. Dantzig, "Letter to the Editor—A Comment on Edie's "Traffic Delays at Toll Booths"," *Journal of the Operations Research Society of America*, vol. 2, no. 3, pp. 339–341, 1954.

[12] A. T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier, "An annotated bibliography of personnel scheduling and rostering," *Annals of Operations Research*, vol. 127, pp. 1–4, 2004.

[13] C. Canon, "Personnel scheduling in the call center industry," *4OR*, vol. 5, no. 1, pp. 89–92, 2007.

[14] L. Bianchi, J. Jarrett, and R. C. Hanumara, "Improving forecasting for telemarketing centers by ARIMA modeling with intervention," *International Journal of Forecasting*, vol. 14, no. 4, pp. 497–504, 1998.

[15] J. W. Taylor, "A comparison of univariate time series methods for forecasting intraday arrivals at a call center," *Management Science*, vol. 54, no. 2, pp. 253–265, 2008.

[16] G. Koole and A. Mandelbaum, "Queueing models of call centers: an introduction," *Annals of Operations Research*, vol. 113, pp. 41–59, 2002.

[17] J. Gong, M. Yu, J. Tang, and M. Li, "Staffing to maximize profit for call centers with impatient and repeat-calling customers," *Mathematical Problems in Engineering*, vol. 2015, Article ID 926504, 10 pages, 2015.

[18] L. Brown, N. Gans, A. Mandelbaum et al., "Statistical analysis of a telephone call center: a queueing-science perspective," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 36–50, 2005.

[19] G. Koole and E. van der Sluis, "Optimal shift scheduling with a global service level constraint," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 35, no. 11, pp. 1049–1055, 2003.

[20] G. M. Thompson, "Labor staffing and scheduling models for controlling service levels," *Naval Research Logistics (NRL)*, vol. 44, no. 8, pp. 719–740, 1997.

[21] 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, pp. 333–358, 2004.

[22] W. B. Henderson and W. L. Berry, "Determining Optimal Shift Schedules for Telephone Traffic Exchange Operators," *Decision Sciences*, vol. 8, no. 1, pp. 239–255, 1977.

[23] T. Aykin, "Optimal shift scheduling with multiple break windows," *Management Science*, vol. 42, no. 4, pp. 591–602, 1996.

[24] T. Aykin, "Comparative evaluation of modeling approaches to the labor shift scheduling problem," *European Journal of Operational Research*, vol. 125, no. 2, pp. 381–397, 2000.

[25] H. K. Alfares, "Operator staffing and scheduling for an IT-help call centre," *European Journal of Industrial Engineering*, vol. 1, no. 4, pp. 414–430, 2007.

[26] E. I. Ásgeirsson and G. L. Sigurðardóttir, "Near-optimal MIP solutions for preference based self-scheduling," *Annals of Operations Research*, vol. 239, no. 1, pp. 273–293, 2016.

[27] 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.

[28] A. N. Avramidis, M. Gendreau, P. L'Ecuyer, and O. Pisacane, "Simulation-based optimization of agent scheduling in multiskill call centers," in *Proceedings of the 2007 5th International Industrial Simulation Conference, ISC 2007*, pp. 255–263, Netherlands, June 2007.

[29] A. N. Avramidis, W. Chan, and P. L'Ecuyer, "Staffing multi-skill call centers via search methods and a performance approximation," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 41, no. 6, pp. 483–497, 2009.

[30] I. Castillo, T. Joro, and Y. Y. Li, "Workforce scheduling with multiple objectives," *European Journal of Operational Research*, vol. 196, no. 1, pp. 162–170, 2009.

[31] 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.

[32] M. Defraeye and I. Van Nieuwenhuyse, "A branch-and-bound algorithm for shift scheduling with stochastic nonstationary demand," *Computers & Operations Research*, vol. 65, pp. 149–162, 2016.

[33] Sergey Dudin, Chesoong Kim, Olga Dudina, and Janghyun Baek, "Queueing System with Heterogeneous Customers as a Model of a Call Center with a Call-Back for Lost Customers," *Mathematical Problems in Engineering*, vol. 2013, Article ID 983723, 13 pages, 2013.

[34] K. Ertogral and B. Bamuqabel, "Developing staff schedules for a bilingual telecommunication call center with flexible workers," *Computers & Industrial Engineering*, vol. 54, no. 1, pp. 118–127, 2008.

[35] M. Excoffier, C. Gicquel, and O. Jouini, "A joint chance-constrained programming approach for call center workforce

scheduling under uncertain call arrival forecasts," *Computers & Industrial Engineering*, vol. 96, pp. 16–30, 2016.

[36] L. V. Green, P. J. Kolesar, and W. Whitt, "Coping with time-varying demand when setting staffing requirements for a service system," *Production Engineering Research and Development*, vol. 16, no. 1, pp. 13–39, 2007.

[37] S. Helber and K. Henken, "Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials," *OR Spectrum*, vol. 32, no. 1, pp. 109–134, 2010.

[38] M. Hojati, "Near-optimal solution to an employee assignment problem with seniority," *Annals of Operations Research*, vol. 181, pp. 539–557, 2010.

[39] R. Ibrahim and P. L'Ecuyer, "Forecasting call center arrivals: Fixed-effects, mixed-effects, and bivariate models," *Manufacturing and Service Operations Management*, vol. 15, no. 1, pp. 72–85, 2013.

[40] A. Ingolfsson, F. Campello, X. Wu, and E. Cabral, "Combining integer programming and the randomization method to schedule employees," *European Journal of Operational Research*, vol. 202, no. 1, pp. 153–163, 2010.

[41] N. Kyngäs, J. Kyngäs, and K. Nurmi, "Optimizing Large-Scale Staff Rostering Instances," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, vol. II, pp. 1524–1531, 2012.

[42] S. Liao, C. Van Delft', G. Koole, Y. Dallery, and O. Jouini, "Call center capacity allocation with random workload," in *Proceedings of the 2009 International Conference on Computers and Industrial Engineering, CIE 2009*, pp. 851–856, France, July 2009.

[43] Z. Liu, Z. Liu, Z. Zhu, Y. Shen, and J. Dong, "Simulated annealing for a multi-level nurse rostering problem in hemodialysis service," *Applied Soft Computing*, vol. 64, pp. 148–160, 2018.

[44] S. Mattia, F. Rossi, M. Servilio, and S. Smriglio, "Staffing and scheduling flexible call centers by two-stage robust optimization," *OMEGA - The International Journal of Management Science*, vol. 72, pp. 25–37, 2017.

[45] D. Millán-Ruiz and J. I. Hidalgo, "Forecasting call centre arrivals," *Journal of Forecasting*, vol. 32, no. 7, pp. 628–638, 2013.

[46] J. E. Nah and S. Kim, "Workforce planning and deployment for a hospital reservation call center with abandonment cost and multiple tasks," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 297–309, 2013.

[47] E. L. Örmeci, F. S. Salman, and E. Yücel, "Staff rostering in call centers providing employee transportation," *OMEGA - The International Journal of Management Science*, vol. 43, pp. 41–53, 2014.

[48] A. Pot, S. Bhulai, and G. Koole, "A simple staffing method for multiskill call centers," *Manufacturing and Service Operations Management*, vol. 10, no. 3, pp. 421–428, 2008.

[49] M. I. Restrepo, B. Gendron, and L.-M. Rousseau, "A two-stage stochastic programming approach for multi-activity tour scheduling," *European Journal of Operational Research*, vol. 262, no. 2, pp. 620–635, 2017.

[50] T. R. Robbins and T. P. Harrison, "A simulation based scheduling model for call centers with uncertain arrival rates," in *Proceedings of the 2008 Winter Simulation Conference, WSC 2008*, pp. 2884–2890, USA, December 2008.

[51] T. R. Robbins and T. P. Harrison, "A stochastic programming model for scheduling call centers with global service level agreements," *European Journal of Operational Research*, vol. 207, no. 3, pp. 1608–1619, 2010.

[52] G. Kilincli Taskiran and X. Zhang, "Mathematical models and solution approach for cross-training staff scheduling at call centers," *Computers & Operations Research*, vol. 87, pp. 258–269, 2017.

[53] 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.

[54] J. Weinberg, L. D. Brown, and J. . Stroud, "Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data," *Journal of the American Statistical Association*, vol. 102, no. 480, pp. 1185–1198, 2007.

[55] F. Rossi, P. Van Beek, and T. Walsh, *Handbook of Constraint Programming*, Elsevier, Amsterdam, Netherland, 2006.

[56] R. Soto, B. Crawford, W. Palma et al., "Top- k Based Adaptive Enumeration in Constraint Programming," *Mathematical Problems in Engineering*, vol. 2015, Article ID 580785, 12 pages, 2015.

[57] L. Trilling, A. Guinet, and D. Le Magny, "Nurse scheduling using integer linear programming and constraint programming," *IFAC Proceedings Volumes*, vol. 39, no. 3, pp. 671–676, 2006.

[58] J.-P. Métivier, P. Boizumault, and S. Loudni, "Solving nurse rostering problems using soft global constraints," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5732, pp. 73–87, 2009.

[59] R. Qu and F. He, "A Hybrid Constraint Programming Approach for Nurse Rostering Problems," in *Applications and Innovations in Intelligent Systems XVI*, pp. 211–224, Springer, London, UK, 2009.

[60] F. He and R. Qu, "A constraint programming based column generation approach to nurse rostering problems," *Computers & Operations Research*, vol. 39, no. 12, pp. 3331–3343, 2012.

[61] E. Rahimian, K. Akartunalı, and J. Levine, "A hybrid integer and constraint programming approach to solve nurse rostering problems," *Computers & Operations Research*, vol. 82, pp. 83–94, 2017.

[62] R. Cipriano, L. Di Gaspero, and A. Dovier, "Hybrid approaches for rostering: A case study in the integration of constraint programming and local search," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 4030, pp. 110–123, Springer, Berlin, Germany, 2006.

[63] S. Bertels and T. Fahle, "A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem," *Computers & Operations Research*, vol. 33, no. 10, pp. 2866–2890, 2006.

[64] T. H. Yunes, A. V. Moura, and C. C. de Souza, "Hybrid column generation approaches for urban transit crew management problems," *Transportation Science*, vol. 39, no. 2, pp. 273–288, 2005.

[65] G. Laporte and G. Pesant, "A general multi-shift scheduling system," *Journal of the Operational Research Society*, vol. 55, no. 11, pp. 1208–1217, 2004.

[66] S. Demassey, G. Pesant, and L.-M. Rousseau, "A Cost-Regular Based Hybrid Column Generation Approach," *Constraints. An International Journal*, vol. 11, no. 4, pp. 315–333, 2006.

[67] M.-C. Côté, B. Gendron, and L.-M. Rousseau, "Modeling the Regular Constraint with Integer Programming," in *Lecture Notes in Computer Science*, vol. 4510, pp. 29–43, Springer, Berlin, Germany, 2007.

[68] M.-C. Côté, B. Gendron, C.-G. Quimper, and L.-M. Rousseau, "Formal languages for integer programming modeling of shift scheduling problems," *Constraints. An International Journal*, vol. 16, no. 1, pp. 54–76, 2011.

[69] N. Chapados, M. Joliveau, and L.-M. Rousseau, "Retail store workforce scheduling by expected operating income maximization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6697, pp. 53–58, 2011.

[70] H. Li and K. Womer, "A decomposition approach for shipboard manpower scheduling," *Military Operations Research*, vol. 14, no. 3, pp. 67–90, 2009.

[71] R. Sahraeian and M. Namakshenas, "On the optimal modeling and evaluation of job shops with a total weighted tardiness objective: constraint programming vs. mixed integer programming," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 39, no. 2, pp. 955–964, 2015.

[72] C. Sel, B. Bilgen, J. M. Bloemhof-Ruwaard, and J. G. A. J. van der Vorst, "Multi-bucket optimization for integrated planning and scheduling in the perishable dairy supply chain," *Computers & Chemical Engineering*, vol. 77, pp. 59–73, 2015.

[73] 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.

[74] A. N. Avramidis, A. Deslauriers, and P. L'Ecuyer, "Modeling daily arrivals to a telephone call center," *Management Science*, vol. 50, no. 7, pp. 896–908, 2004.

[75] S. Bhulai, "Dynamic routing policies for multiskill call centers," *Probability in the Engineering and Informational Sciences*, vol. 23, no. 1, pp. 101–119, 2009.

[76] T. Aktekin, "Call center service process analysis: Bayesian parametric and semi-parametric mixture modeling," *European Journal of Operational Research*, vol. 234, no. 3, pp. 709–719, 2014.

[77] A. Sencer and B. Basarir Ozel, "A simulation-based decision support system for workforce management in call centers," *Simulation*, vol. 89, no. 4, pp. 481–497, 2013.

[78] "Detailed Scheduling in IBM ILOG CPLEX Optimization Studio with IBM ILOG CPLEX CP Optimizer," *Ibm*, 2010. Accessed: 11-Jan-2016. Available: https://www.researchgate.net/publication/328530729_Detailed_Scheduling_in_IBM_ILOG_CPLEX_Optimization_Studio.

[79] J.-C. Régin, "Solving Problems with CP: Four Common Pitfalls to Avoid," in *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming - CP 2011*, pp. 3–11, Springer, Berlin, Germany, 2011.

Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

The Scientific
World Journal

Journal of
Probability and Statistics

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Optimization

International Journal of
Engineering
Mathematics

International Journal of
Analysis

Hindawi

Submit your manuscripts at
www.hindawi.com

Journal of
Complex Analysis

Advances in
Numerical Analysis

Mathematical Problems
in Engineering

International Journal of
Differential Equations

Discrete Dynamics in
Nature and Society

International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces

Abstract and
Applied Analysis

Advances in
Mathematical Physics