

## Research Article

# The Air Traffic Controller Work-Shift Scheduling Problem in Spain from a Multiobjective Perspective: A Metaheuristic and Regular Expression-Based Approach

Faustino Tello, Alfonso Mateos , Antonio Jiménez-Martín , and Adán Suárez

Decision Analysis and Statistics Group, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Madrid, Spain

Correspondence should be addressed to Antonio Jiménez-Martin; antonio.jimenez@upm.es

Received 23 June 2017; Revised 15 December 2017; Accepted 8 January 2018; Published 11 February 2018

Academic Editor: Danielle Morais

Copyright © 2018 Faustino Tello et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We address an air traffic control operator (ATCo) work-shift scheduling problem. We consider a multiple objective perspective where the number of ATCos is fixed in advance and a set of ATCo labor conditions have to be satisfied. The objectives deal with the ATCo work and rest periods and positions, the structure of the solution, the number of control center changes, or the distribution of the ATCo workloads. We propose a three-phase problem-solving methodology. In the first phase, a heuristic is used to derive infeasible initial solutions on the basis of templates. Then, a multiple independent run of the simulated annealing metaheuristic is conducted aimed at reaching feasible solutions in the second phase. Finally, a multiple independent simulated annealing run is again conducted from the initial feasible solutions to optimize the objective functions. To do this, we transform the multiple to single optimization problem by using the rank-order centroid function. In the search processes in phases 2 and 3, we use regular expressions to check the ATCo labor conditions in the visited solutions. This provides high testing speed. The proposed approach is illustrated using a real example, and the optimal solution which is reached outperforms an existing template-based reference solution.

## 1. Introduction

The core of air traffic control operator (ATCo) activity is to facilitate airspace and airport surface traffic flow, while avoiding collisions between aircraft. To satisfy this essential safety constraint, they must detect and solve possible conflicts between trajectories.

As a human ATCo can only handle a limited amount of traffic, the airspace is divided into a number of sectors. These sectors are operated by two ATCo working positions (executive and planner). All the sectors open at any one time are referred to as *sectorization*. The sectorization changes throughout the day depending on the air traffic.

The sectorization required to handle the estimated traffic for a time period can be designed beforehand. Therefore, a very important problem in air traffic control is to determine the minimum number of ATCos necessary to cover a sectorization structure for a given time period, denoted as

*airspace sector configuration*, while satisfying certain ATCo labor conditions, including, for instance, resting and working time distributions. Alternatively, the number of ATCos could be fixed. The aim then would be to distribute ATCos to cover the corresponding *airspace sector configuration*.

These optimization problems belong to the class of *timetabling and scheduling problems*. The size and complexity of these combinatorial problems make them hard or even impossible to solve with exact methods.

Different problem-solving approaches have been proposed in the literature to deal with timetabling problems [1]. The Third International Timetabling Competition (ITC2011) [2] motivated the development of several approaches for the extended *markup language for high school timetabling* (XHSTT) problem [3]. The four finalists employed metaheuristics as part of or as the main problem solver [4–7].

Recent problem-solving approaches for the XHSTT problem are based on variable neighborhood search (VNS) [8],

simulated annealing (SA) [9], or *matheuristics* (the integration of metaheuristics and mathematical programming) [10, 11].

Different approaches can also be found in the literature concerning other timetabling problems, including operational research, metaheuristics, or novel intelligent methods, such as university course timetabling problems [12], job shop scheduling problems [13–15], and sports scheduling problems [16].

Regarding work-shift scheduling problems in the context of air traffic management (ATM), shiftwork management is addressed in [17], which contains a literature review about the impact of shiftwork, its consequences for health, safety, productivity, and efficiency, as well as social implications. It also refers to the EC Directive 93/104 that establishes the European regulation for working time design in ATM.

A study on shiftwork practices in both ATM and other industries, such as medical, police, and airline industries, is presented in [18]. It concludes that although there are a range of software tools, in many cases involving ATM, they are costly and not completely suited to the needs. The strengths and weaknesses of automated scheduling tools have already been outlined [19].

More recently, Stojadinovic [20] gives an in-depth description of three problem encodings capable of formulating a very broad set of different scheduling requirements considering a time period of a month or a year. The problem was solved using propositional satisfiability (SAT, [21]), MaxSAT, the pseudo-Boolean, satisfiability modulo theory, constraint satisfaction, and integer linear problem solvers. In combination with these solvers, three different optimization techniques were presented. Results suggest that SAT-related approaches outperform other problem-solving methods.

Later, Stojadinovic [22] presents a combination of the SAT problem-solving and the hill climbing method. First, the SAT solver is used to generate a feasible solution. Then, hill climbing is used to improve this solution in terms of the number of satisfied ATCo demands. Finally, SAT problem-solving is used to further improve the identified solution by fixing some parts of the solution. The process is repeated until an optimal solution is found.

A simplified version of the ATCo work-shift scheduling problem considered in this paper was previously solved. This solution considered a single core, accounting for only one type of sector. Consequently, all available ATCos were able to operate in all sectors. Another aim was to minimize the number of ATCos required to cover a given airspace sector configuration, while satisfying a set of ATCo working conditions. Additionally, the different shifts per day were simultaneously optimized.

In this paper, however, we consider the possibility there being more than one core with common sectors, taking into account the en route and approach sectors and ATCos with different operating credentials. The number of available ATCos to cover a given airspace sector configuration is now known in advance, and only one shift is optimized. Besides, we consider a multiobjective perspective, accounting for issues such as some conditions with respect to ATCo work and rest periods and positions, the structure of the solution,

the number of control center changes, or the distribution of the ATCo workloads.

The problem-solving methodology used in both papers is similar in the sense that a heuristic is first proposed to build different initial solutions based on the use of an optimized template and the modification of rest period lengths and then, multiple independent runs of the simulated annealing (SA) metaheuristic are used to reach the optimal solution.

However, the heuristic in the simplified version derives different feasible solutions, whereas the new heuristic presented in this paper accounts for the sector types and ATCo credentials and derives infeasible initial solutions (more than the number of available ATCos may be used and some working conditions are violated). Hence, simulated annealing is conducted based on a multiple independent run algorithm aimed at reaching feasible solutions from the infeasible solutions derived by the heuristic. Then, a second multiple independent run of SA is conducted to reach the optimal solutions accounting for the different objectives functions mentioned above.

Note that regular expressions (Regex) [23] are used in both search processes. A Regex is a sequence of characters that define a search pattern that is matched to the string representation of every solution. In our context, the patterns represent the violation of ATCo working conditions. The benefits of using Regex are high testing speed and modularity for a clear and maintainable implementation of the optimization model.

The paper is structured as follows. Section 2 describes the ATCo work-shift scheduling problem, together with the constraints accounting for ATCo working conditions and the objective functions under consideration. The proposed problem-solving methodology together with regular expressions and their application for checking working conditions are described in Section 3. An example is used to illustrate the proposed methodology in Section 4. Finally, some conclusions are outlined in Section 5.

## 2. Problem Description

One of the core tasks of ATCos is to avoid collisions between aircraft. To do this, the ATCo must assure that the aircraft are always separated by a minimum safety distance, denoted as *separation standards*.

The airspace is composed of volumes (elementary unit of airspace). As a human ATCo can only handle a limited amount of traffic, the airspace is divided into a number of sectors. These sectors may include one or more volumes. All the sectors open at any one time must cover all the volumes of the corresponding airspace. This is referred to as *airspace sector configuration*.

Additionally, a *core* consists of a set of sectors, and a sector may belong to several cores. A control center may be responsible for managing one or more cores, depending on the control center under consideration. If more than one core have sectors in common, then the ATCo assignment process for the respective cores should be simultaneous. Otherwise, each core should be solved separately.

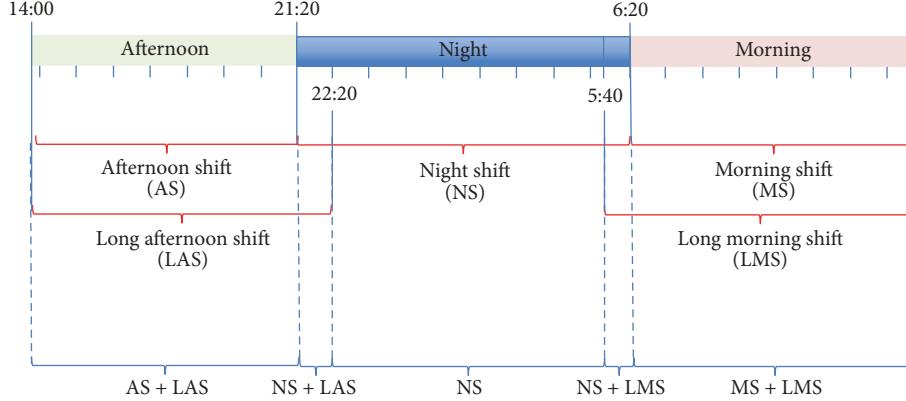


FIGURE 1: Daily working shifts in Canary Islands control center.

Two sectors are called *related sectors* if they share a volume. As explained later, the existence of related sectors plays an important role in some working conditions and an affinity matrix specifying which sectors are related to each other will be an optimization problem input. Moreover, sectors can be categorized as *approach* and *en route* sectors. Approach sectors are generally 5 to 10 nautical miles (9 to 18 km) from the airport depending on the airport procedures, whereas *en route* sectors are further away.

Each sector is operated by two ATCos, the executive and the planner ATCos. The executive ATCo talks to the aircraft and gives instructions to the pilots to avoid conflict situations between aircraft, whereas the planner ATCo is responsible for anticipating possible conflicts between aircraft and communicating with the executive ATCo the problem before it materializes. ATCos are qualified to operate on a particular core and can be categorized as PTD or CON ATCo depending on the type of sector for which they are accredited. A PTD ATCo can operate *en route* and approach sectors, whereas a CON ATCo can only operate *en route* sectors.

The sector configuration changes throughout the day depending on the air traffic. A higher volume of air traffic means more sectors with smaller dimensions will be opened, thus requiring more ATCos. As a result, sectors are dynamically divided and merged over time depending on the air traffic, and the number of ATCos necessary to cover the open sectors varies accordingly.

The sector configurations needed to handle the estimated traffic for a time period (usually a day) can be designed beforehand. This is denoted as *airspace sector configuration*.

Therefore, a very important problem in air traffic control could be to determine the minimum number of ATCos necessary to cover an airspace sector configuration for a given time period, while satisfying certain strong constraints accounting for ATCo working conditions, including, for instance, resting and working time distributions.

In this paper, however, we consider that the number of available ATCos is fixed in advance and the problem is to cover the given airspace sector configuration taking into account working conditions but accounting for the optimization of the ATCo work and rest periods and positions,

the structure of the solution, the number of control center changes, or the distribution of the ATCo workloads.

Besides, ATCos can work different shifts. Shift lengths may vary at different control centers, and some may not even be considered. Figure 1 shows the five ATCo shifts used at Canary Islands: long morning (LMS) (6:30–15:00 h.), morning (MS) (7:30–15:00), afternoon (AS) (15:00–22:00), long afternoon (LAS) (15:00–23:00), and night (NS) (22:00–7:30).

Note that ATCos working AS and LAS shifts simultaneously work in the afternoon, as shown in Figure 1. The same applies for ATCos working NS and LAS shifts in the time period (22:00–23:00), for ATCos working NS and LMS shifts in the time period (6:30–7:30), and for ATCos working MS and LMS shifts in the morning. In the time period (23:00–6:30), only the NS ATCos are working.

The stated problem will address the optimization of the morning, afternoon, or night shifts. With regard to the optimization of the morning shifts, we account for MS and LMS ATCos and, consequently, some sectors belonging to the end of the previous night will be assigned to LMS ATCos. With regard to the optimization of the afternoon shifts, we account for AS and LAS ATCos and, consequently, some sectors belonging to the beginning of the following night will be assigned to LMS ATCos.

However, when we optimize the night shift, only NS ATCos are considered, and sectors (at the beginning) assigned to LAS ATCos when optimizing the previous afternoon and sectors (at the end) assigned to LMS ATCos when optimizing the following morning are removed from the analysis.

Figure 2 shows an example of an airspace sector configuration for a night in Spain's Canary Islands, considering two cores. Three sectors are always open during the night shift, but the total number of sectors involved in the airspace sector configuration is 5. Specifically, the sector GCCCOCE is opened throughout the shift, sectors GCCAAC and GCCRE2 are open at the beginning (until 23:00 and 00:00, resp.) and at the end (from 6:45 in both cases) of the shift, and sectors GCCCTM3 and GCCCRU6 are open in the middle of the shift (from 0:00 and 23:00, to 6:45).

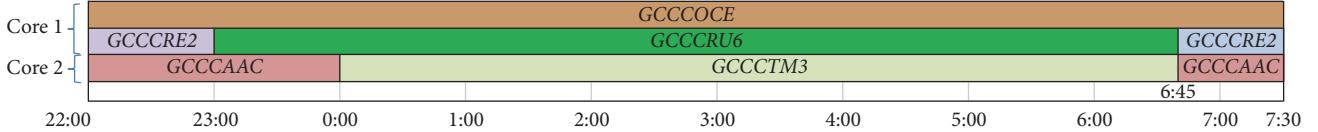


FIGURE 2: Canary Islands airspace sector configuration.

As already mentioned, a number of constraints accounting for ATCo working conditions have to be taken into account. In Spain, all these conditions were compiled and published in the Official State gazette (Boletín Oficial del Estado, BOE), Royal Decree 1001/2010, and Law 9/2010, regulating the provision of air traffic services. ATCo working conditions are as follows:

- (1) (LC1) ATCos can only operate sectors belonging to the core for which they are qualified.
- (2) (LC2) CON ATCos can only operate en route sectors.
- (3) (LC3) ATCos must rest for 25% of the work shift in day shifts (MS, LMS, AS, and LAS). The night shift (NS) rest periods account for 33% of working time.
- (4) (LC4) A sector opened during the entire night shift must be operated by four ATCos.
- (5) (LC5) ATCos cannot work for more than two consecutive hours.
- (6) (LC6) ATCos can only operate during one shift.
- (7) (LC7) ATCos should rest at least half an hour every two hours.
- (8) (LC8) Sector changes are not allowed without resting unless there is an emergency. ATCos can change to a related sector without resting.
- (9) (LC9) The minimum work period length of a controller between two breaks is 15 minutes.
- (10) (LC10) Each rest period should last at least 15 minutes.
- (11) (LC11) ATCos must remain in the same sector and position for at least 15 minutes (minimum time in a position).
- (12) (LC12) Each ATCo can work in at most three different nonrelated sectors in the respective shift.
- (13) (LC13) Each work shift must have one ATCo assigned. And each ATCo must be responsible for one shift.
- (14) (LC14) Each ATCo must work at least 15 minutes and cannot rest all the shift.

As pointed out previously, we will account for several objectives in the ATCo work-shift scheduling problem under consideration.

First, the ATCo work and rest periods and positions should be as close as possible to fixed values. Specifically, the time that ATCos remain in the same sector and working position (planner or executive) should be as close as possible to 45 minutes, the working time between two rest periods should be as close as possible to 90 minutes, and the

percentage of time ATCos work an executive position must be between 40% and 60% of the total working time (excluding rest periods).

Secondly, the problem-solving structure should be similar to previous template-based solution. This makes the solution easier to understand for the control center staff and would facilitate possible hand-made changes. Additionally, the number of control center changes should be minimized. To do this, the number of rest periods should be minimized. Finally, a balanced ATCo workload distribution is desirable.

Note that ordinal information regarding the relative importance of the above objectives was provided by experts from the Reference Center for Research, Development and Innovation in ATM (CRIDA, <http://www.crida.es>), a non-profit joint venture between ENAIRE, responsible for the air traffic management and control and for communication, navigation, and surveillance networks in Spain, Universidad Politécnica de Madrid, and Ineco, a global leader in infrastructure engineering and consultancy. The order of importance of the objectives matches up with the order in which they have been introduced.

In summary, the information necessary (inputs) to solve the ATCo work-shift scheduling problem under consideration is the number of cores managed by the control center together with the lists of sectors that they control specifying the type of sector (approach or en route), a binary affinity matrix, the airspace sector configuration established for each core, the shift to be optimized (morning, afternoon, or night), and the number of available PTD and CON ATCos, together with their shifts morning (MS or LMS) or afternoon (AS or LAS) shift optimization.

### 3. Problem-Solving Methodology

In this section, we introduce a methodology to solve the ATCo work-shift scheduling problem under consideration. The proposed methodology is a three-phase process. In the first phase, a heuristic is proposed to build different initial infeasible solutions based on the use of an optimized template and the modification of the rest period lengths. These initial solutions violate some labor conditions and, even worse, possibly use more than the available number of ATCos.

In the second phase, a multiple independent run (MIR) algorithm based on simulated annealing (SA) is applied to the infeasible solutions achieved in the first phase, in order to yield feasible solutions. To do this, we apply a fitness function to try to reduce the number of ATCos used until the number of available ATCos is met while penalizing the number of times labor conditions is violated. Note that the first two phases should be performed as fast as possible since it is

FIGURE 3: Solution representation.

important to know as soon as possible whether or not there is a feasible solution for the available number of ATCos.

In the third phase, a SA-based MIR is again applied starting from the feasible solutions output in the second phase. The aim now is to optimize the objective functions described in the previous section taking into account the ordinal information about their relative importance provided by experts from CRIDA. The original multiobjective optimization problem is then transformed into a single weighted optimization problem, whose weights are derived from the centroid function, which accounts for the available ordinal information.

Note that regular expressions (Regex or Regexp) are used throughout the second and third phases to check most of the constraints associated with working conditions. The benefits of using RegExps are high testing speed and modularity for a clear and maintainable implementation of the optimization model.

Before describing the above phases in detail, let us first examine how the proposed methodology represents the solutions. We use a matrix where the number of columns is the number of time slots in the shift (morning, afternoon, or night) for optimization, and each row is associated with an ATCo. A slot is equivalent to five minutes. The number of rows is established when applying the heuristic to build an initial solution.

Note that initial infeasible solutions and solutions visited throughout the second phase may have different numbers of ATCos. However, the number of rows in the solutions in the third phase will always match up with number of available ATCos.

Each matrix element  $(i, j)$  refers to the state of the ATCo  $i$  in the time slot  $j$  and is represented by three letters. We use the 111 combination to represent a resting ATCo. Otherwise, a combination of three letters starting with  $aaa$  identifies the sector that the ACT is operating in the respective time slot. Uppercase letters are used to signify that the ATCo is working as an executive ATCo in the respective sector, whereas lowercase letters are used for planner positions.

Colors are used to represent sectors when displaying solutions; see Figure 3. Rest periods and out-of-shift periods are in white. Figure 3 is a representation of the airspace sector configuration shown in Figure 2. It consists of 114 slots (columns), nine and a half hours, and it is covered by 14 ATCos (rows).

**3.1. A Heuristic for the Construction of Initial Solutions.** We propose a heuristic to build a set of initial solutions with different rest periods. They will be used afterwards in the MIR algorithm based on SA to reach feasible solutions. The proposed heuristic is based on the use of an optimized template (see Figure 4), with three ATCos covering a sector during 96 time slots (eight hours) without specifying the ATCo positions (executive or planner), which are assigned afterwards.

The duration of the work periods is twice as long as for the rest periods in the template. However, if different rest period durations are used, we have different templates, and the heuristic will build different initial solutions, albeit with a similar structure. As the working conditions establish that each rest period should last at least 15 minutes (three slots), and the maximum work period is two hours (24 slots), the duration of a rest period must be between three and 12 slots. Thus, the heuristic builds ten different initial solutions.

The template in Figure 4 corresponds to an eight-hour shift (96 slots) and a six-slot rest period, where R and W represent rest and work periods, respectively, and lower and uppercase case letters differentiate the working positions.

We denote the number of columns in the solution matrix, that is, the number of slots in the shift to be optimized by  $n_{\text{Col}}$ . The steps of the heuristic are as follows.

*Step 1* (add templates to cover the airspace sector configuration). For each open sector  $i \in \{1, \dots, s\}$  do the following:

- (1) Insert an empty template in the solution matrix.
  - (2) For each time slot  $j \in \{1, \dots, n\text{Col}\}$  do the following:
    - (a) If sector  $i$  is open during the time slot  $j$ , then enter the slot corresponding to the template in the solution matrix in position  $j$  with the name of sector  $j$ .
    - (b) If sector  $i$  is not opened during the time slot  $j$ , then enter resting slots in the solution matrix in position  $j$ .

At the end of Step 1, all the sectors are assigned a pair of ATCos, for which the necessary templates have been inserted. However, certain ATCo working conditions may be violated. We denote the number of rows in the solution matrix, corresponding to the number of ATCos considered, by  $nRow$ . Step 2 is aimed at improving the feasibility of the solution matrix.

ACTCo 1	W	R	w	W	R	w	W	R	w	W	R	w	W	R	w	W
ACTCo 2	R	w	W	R	w	W	R	w	W	R	w	W	R	w	W	R
ACTCo 3	w	W	R	w	W	R	w	W	R	t	T	R	w	W	R	w

w	w	w	w	w	w	w	W	W	W	W	W	W	W	W	W
W	W	W	W	W	W	W	R	R	R	R	R	R	R	R	R

FIGURE 4: Template.

*Step 2* (repair the solution to improve feasibility). For each row  $l \in \{1, \dots, n_{\text{Rows}}\}$ , in the solution matrix that does not meet a minimum work period length, we try to assign the work period whose length is not the minimum to another ATCo. To do this, do the following:

- (1) Try to transfer a work period without the minimum length from one ATCo to another.

- (a) Check if there is another ATCo working in the same sector who starts at the start of the work period to be transferred (this would be ideal since the ATCo would continue to work in the same sector) and the length of that rest period is greater than the respective work period. Then, check if the following constraints are met before making the transfer:
  - (i) ATCos have to rest for 30 minutes every two working hours.
  - (ii) The maximum continuous work period is two hours.
  - (iii) The minimum continuous rest period is 15 minutes.
  - (iv) The minimum continuous work period is 15 minutes.

- (b) Check if there is another ATCo who starts working in that same sector just when the work period to be transferred ends and has a rest period with a length greater than the fragment to be assigned. Then, as in (a), check the above conditions before making the transfer.

- (2) Try to extend a work period without the minimum length using a work period from another ATCo:

- (a) Check if there is another ATCo working in the same sector who started at the same time as the work period began (this would be ideal since the ATCo would be working in the same sector) and has enough slots to complete the minimum working time. Before making the handover, check that it does not imply increasing the infeasibility.
- (b) Check if there is another ATCo who started working in that same sector just after the work

period fragment that we want to expand and has enough working slots to complete the minimum working time.

- (c) Check the list of constraints that the ATCos must meet.

*Step 3* (allocation of available resources). For each row  $l \in \{1, \dots, n_{\text{ATCo}}\}$ , in the solution matrix do the following:

- (1) If row  $l$  contains approach sectors during the long shift, then we search for an available PTD ATCo associated with the long shift who can operate in that core. This ATCo is assigned to row  $l$ .
- (2) If row  $l$  contains approach sectors during the short shift, then search for an available PTD ATCo associated with the short shift who can operate in that core. This ATCo is assigned to row  $l$ . Otherwise, we search for an available PTD ATCo associated with the long shift who can operate in that core and assign this ATCo to that row.
- (3) If row  $l$  contains en route sectors during the long shift, then we search for an available CON ATCo associated with the long shift who can operate in that core. This ATCo is assigned to row  $l$ . Otherwise, we search for an available PTD ATCo associated with the long shift who can operate in that core and assign this ATCo to that row.
- (4) If row  $l$  contains en route sectors during the short shift, then we search for an available CON ATCo associated with the short shift who can operate in that core. This ATCo is assigned to row  $l$ . Otherwise, we search for an available CON ATCo associated with the long shift, an available PTD ATCo associated with the short shift, or an available PTD ATCo associated with the long shift (following that order) who can operate in that core and assign this ATCo to that row.

At this point, some ATCos may not have been assigned since they do not meet the requirements per row. These ATCs are assigned as follows.

Check if each ATCo  $c \in \{1, \dots, n_{\text{ATCo}}\}$  is assigned to a row in the solution matrix.

If ATCo  $c$  has not been assigned and is a COM (PTD) ATCo, then we search for a row including en route (approach) sectors.

- (1) If there is an available row including en route (approach) sectors, then this row will be assigned to ATCo  $c$ .
- (2) Otherwise, a random available row is assigned to ATCo  $c$ .

Once all the available ATCos have been assigned, there may still be rows without an assigned ATCo. In this case, we add artificial ATCos; that is, the initial solution includes more ATCos than available. These artificial ATCos will be removed in the second phase of the problem-solving method, leading to feasible solutions.

There is an exception to the application of the above heuristic for night shifts. If there is an open sector throughout the entire night shift, then a template including four ATCos with rest period lengths of 9 slots (45 minutes) is used to cover the sector. Note that these ATCos will only work in that sector. Consequently, we would have a lower dimensional problem (sectors open throughout the night shift are already covered) where there are fewer available ATCos (some of the originally available ATCos have been assigned to the above sector).

**3.2. A MIR Algorithm Based on SA to Reach Feasible Solutions.** The MIR algorithm based on SA starts with the infeasible solutions output in the first phase with the aim of reaching feasible solutions.

First, we will briefly describe the simulated annealing metaheuristic, which is used in both this and the third phase. We discuss how it is adapted to the problems addressed in both phases later.

*Simulated annealing (SA)* [24, 25] is a trajectory-based metaheuristic which is named for and inspired by annealing in metallurgy. The basic idea of SA is as follows. An initial feasible solution,  $x_0$ , is randomly generated. Then, in each iteration  $i$ , a new solution ( $y$ ) is randomly generated from the neighborhood,  $N(x_i)$ , of the solution considered in that iteration,  $x_i$ . If the new solution is better than the current one, then the algorithm moves to that solution. Otherwise, there is some probability of it moving to a worse solution. The acceptance of worse solutions makes for a broader search for the optimal solution and avoids trapping in local optima in early iterations.

The search is initially very much diversified, since practically all moves are allowed. As the temperature drops, the probability of accepting a worse moves decreases, and only better moves will be accepted when it is zero. This makes SA work like hill climbing.

SA pseudocode for a minimization optimization problem is as follows.

*Initialization.* Do  $x^* = x_0$ ,  $f^* = f(x_0)$ ,  $i = 0$ . Select the initial temperature  $t_0 = T$  ( $t_i$ : temperature in step  $i$ ).

Repeat until stopping criterion is satisfied:

Randomly generate  $y_i \in N(x_i)$

If  $f(y_i) - f(x_i) \leq 0$ , then

$$x_{i+1} = y_i$$

If  $f(x^*) > f(y_i)$ , then  $x^* = y_i$ ,  $f^* = f(y_i)$

Else

$$p \sim U(0, 1)$$

If  $p \leq e^{-(f(y_i) - f(x_i))/t_i}$ , then  $x_{i+1} = y_i$

Else  $x_{i+1} = x_i$

Update temperature,  $i = i + 1$

Some elements in the above algorithm require clarification. Generally, the *initial temperature* is set such that the acceptance ratio of worse moves is equal to a certain specified value [26].

The *cooling schedule* defines the way in which the temperature decreases across the iterations. A common cooling schedule is for the temperature to be kept constant for a number of iterations ( $L$ ) and then be decreased it according to a geometric schedule:  $T_k = \alpha^k T_0$ , the typical value for  $\alpha$  being 0.95 [27].

The most commonly used *stopping criteria* is to stop when the improvement in the fitness function is less than a given percentage during a fixed number of iterations.

The SA algorithm is adapted to deal with the search for feasible solutions in the second phase of the proposed methodology as follows: the *initial solutions* in the multiple independent runs of SA are the infeasible solutions output by the heuristic in the first phase.

The aim of the fitness function in this phase is to reduce the number of ATCos used until it matches the number of available ATCos while penalizing the working conditions violated. To do this, we consider the maximization of two weighted components ( $f$  and  $g$ ) in the fitness function as follows:

$$\max f$$

$$= \begin{cases} w_1 * h + w_2 * g, & \text{if } n\text{ATCo} > n\text{ATCoaval} \\ g, & \text{if } n\text{ATCo} \leq n\text{ATCoaval}, \end{cases} \quad (1)$$

where  $w_1$  and  $w_2$  represent the relative importance of the components,  $n\text{ATCo}$  is the number of ATCos used in the current solution and  $n\text{ATCoaval}$  is the number of available ATCos.

$h$  is the component aimed at reducing the number of ATCos:

$$h = \frac{1}{n\text{ATCo}^2} \sum_{i=1}^{n\text{ATCo}} ih_i, \quad (2)$$

where  $h_i$  is the number of slots that the  $i$ th ATCo is working in.

Note that all visited solutions in the search process are reorganized as follows: the ATCo with the smallest workload is placed on the first row of the solution codification, the ATCo with the second smallest workload is placed on the second row, and so on, where the ATCos with the highest workloads are placed on the last rows; that is, the solution matrix is changed by row permutation.

The term  $1/n\text{ATCo}^2$  implies that when the number of ATCos decreases, the fitness value is greatly improved. For

a given number of ATCos, fitness values are higher for solutions with a high workload for ATCos with higher indexes (in the bottom rows of the solution matrix).

Thus, if we take into account the considered fitness function and the row reorganization of the visited solutions, we find that the search process tends to reallocate work periods from ATCos in the top rows to ATCos in the bottom rows (with the highest workload). In this way, we increase the rest periods of the first ATCos and tend to decrease the number of ATCos needed to cover the airspace sector configuration.

In order to correctly use the weighted fitness function, we must normalize the values of this component:

$$h = 1 - \frac{h_{\max} - h}{h_{\max} - h_{\min}}, \quad (3)$$

where

$$h_{\max} = \sum_{k=nATCo^*}^{nATCo} \frac{k \times nSlot}{nATCo^2}, \quad (4)$$

and  $nATCo^*$  is the number of ATCos needed to cover the airspace sector configuration without resting and  $nSlot$  is the number of slots in the considered shift (number of columns in the matrix solution), whereas

$$h_{\min} = \sum_{k=1}^{nATCo^*} \frac{k \times nSlot}{nATCo^2}, \quad (5)$$

associated with a solution where the workload is shared among in the ATCos located in the top rows of the solution matrix, whereas no workload is assigned to ATCos in the bottom rows.

$g$  is the component representing the number of times working conditions are violated. To normalize its values, we proceed as follows:

$$g = \frac{g_{\max} - g}{g_{\max}}, \quad (6)$$

where  $g_{\max}$  is the maximum of number times that the working conditions can be violated, and its value is different depending on the shift that we are optimizing. If we are optimizing a night shift, then

$$\begin{aligned} g_{\max} &= 8 \times nATCo + 2 \times 2nATCo \\ &\quad + 4 \left( nATCo \times +nATCo \times nSlot \times \frac{1}{nSlot} \right) \\ &= 12 \times nATCo \\ &\quad + 4 \left( nATCo \times +nATCo \times \frac{nSlot}{nSlot} \right) \\ &= 20 \times nATCo, \end{aligned} \quad (7)$$

since there are eight constraints (LC3, LC5, LC6, LC9, LC10, LC11, LC12, and LC14) not met by  $nATCo$  controllers, two constraints (LC4 and LC13) not met by  $2 \times nATCo$ , and

four constraints (LC1, LC2, LC7, and LC8) not met by  $nATCo$ , whose degree of default is accounted for and every noncompliance for each slot in the shift is multiplied by  $1/nSlot$ .

If a morning or afternoon shift is being optimized, then LC4 is not considered and

$$g_{\max} = 10 \times nATCo + 4 (nATCo + nATCo). \quad (8)$$

Regarding the *neighborhood structure*, the process for randomly selecting a solution in the neighborhood of a given solution is as follows:

- (1) An ATCo is selected randomly. The aim is to reallocate part of this ATCo's workload to another ATCo.
- (2) A work period of the chosen ATCo is selected randomly. Then, a part of that work period is considered for reallocation. The number of slots containing this part is randomly chosen from possible 3, 6, 9, and 12; that is, if the work period length is 10 or 11 slots then only the values 3, 6, and 9 are used and so on.
- (3) A second ATCo is selected at random and we check if the whole work period for reallocation can be assigned to that ATCo. If this is not possible, then we try with another ATCo and so on. To do this, we check that the ATCo under consideration is resting in the slots corresponding to the work period to be reallocated.

If it is not possible to reallocate the considered work period to any ATCo, then we again randomly select the number of slots in the part of the work period for reallocation, considering lower candidate values lower than the one we used previously and repeat the process.

We repeat the process until the work period is reallocated or all four possible piece sizes (3, 6, 9, and 12) have been considered, in which case we go back to Step (1).

As mentioned above, the first two phases should be performed as fast as possible since it is important to know as soon as possible whether or not there is a feasible solution for the available number of ATCos.

**3.3. A MIR Algorithm Based on SA to Reach the Optimal Solution.** In the third phase, a MIR of SA is again conducted starting from the feasible solutions output in the second phase. The aim now is to optimize four objective functions taking into account the ordinal information about their relative importance provided by experts from CRIDA.

First, the ATCo work and rest periods and positions should be as close as possible to fixed values. Specifically, the time ATCos remain in the same sector and working position (planner or executive) should be as close as possible to 45 minutes, the working time between two rest periods should be as close as possible to 90 minutes, and the percentage of time ATCos work in executive positions must be between 40% and 60% of the total working time (excluding rest periods).

Second, the solution structure should be similar to the previous template-based solution. This makes the solution more understandable for the control center staff and would facilitate any possible hand-made changes.

Additionally, the number of control center changes should be minimized. To do this, the number of rest periods should be minimized. Finally, the ATCo workload distribution should be balanced.

The original multiobjective optimization problem is then transformed into a single weighted optimization problem, whose weights are derived from the rank-order centroid (ROC) method, which accounts for the available ordinal information provided by CRIDA experts. The ROC method has an appealing theoretical rationale and appears to perform better than the other rank-based schemes in terms of choice accuracy [28].

Accordingly to the ROC method,

$$w_i = \frac{\sum_{j=i}^n 1/j}{n}, \quad i = 1, \dots, n; \quad (9)$$

hence

$$\begin{aligned} w_1 &= \frac{1 + 1/2 + 1/3 + 1/4}{4} = 0.52, \\ w_2 &= \frac{1/2 + 1/3 + 1/4}{4} = 0.27, \\ w_3 &= \frac{1/3 + 1/4}{4} = 0.15, \\ w_4 &= \frac{1/4}{4} = 0.06. \end{aligned} \quad (10)$$

As mentioned above, the number of rows in the solution matrices will always be the same in the third phase, representing the number of available ATCos, since the solutions (initial and visited throughout the search) are feasible. Besides, the permutation process applied to all solutions (initial and visited) in the second phase will not be necessary in the third phase since a different fitness function is considered.

Now, we describe how the different objective functions have been modeled.

**3.3.1. Objective 1: Desirable ATCo Work and Rest Periods and Positions.** This objective accounts for three equally important subobjectives.

(1) *The Time ATCos Remain in the Same Sector and Working Position (Planner or Executive) Should Be as Close as Possible to 45 Minutes.* The procedure for checking this point is as follows: for each row in the solution matrix, we compute the number of consecutive slots during which the ATCo remains in the same sector and work position and the difference between this value and the optimum value (9 slots = 45 minutes). Then, we add up all the differences for the different work periods in the row and again for all the rows in the

solution matrix. Finally, we compute the mean value by dividing the above amount by the number of ATCos:

$$f_{1,1} = \frac{\sum_{i=1}^{n\text{ATCo}} \sum_{j=1}^{n_i} |45 - l_{i,j}|}{n\text{ATCo}}, \quad (11)$$

where  $n_i$  is the number of work periods during which the  $i$ th ATCo remains in the same sector and working position and  $l_{i,j}$  is their duration in minutes.

This objective function is then normalized:

$$g_{1,1} = \frac{f_{1,1}^{\max} - f_{1,1}}{f_{1,1}^{\max}}, \quad (12)$$

where  $f_{1,1}^{\max} = |45 - \text{posMin}| \times 8 \times (n\text{Slot}/30)$ , posMin is the minimum time an ATCo can remain in a position without violating any working condition, and  $n\text{Slot}/30$  is the number of times; there is a two-hour work period (24 slots) followed by a 30-minute (6 slots) rest period in the shift.

(2) *The Optimal Working Time between Breaks Should Be 90 Minutes.* Now, we compute for each row the length of each work period (in minutes) and its difference from the optimum work period (90 minutes). We add these differences for the row and then for all rows. Finally, we divide the result by the number of rows (ATCos) to derive the mean value:

$$f_{1,2} = \frac{\sum_{i=1}^{n\text{ATCo}} \sum_{j=1}^{m_i} |90 - r_{i,j}|}{n\text{ATCo}}, \quad (13)$$

where  $m_i$  is the number of work periods of ATCo  $i$  and  $r_{i,j}$  is their duration in minutes.

Now, we proceed to normalization:

$$g_{1,2} = \frac{f_{1,2}^{\max} - f_{1,2}}{f_{1,2}^{\max}}, \quad (14)$$

where  $f_{1,2}^{\max} = |90 - \text{workMin}| \times (n\text{Slot}/6)$  is the maximum value of  $f_{1,2}$ , with workMin being the minimum working time for an ATCo (15 minutes). We divide the number of slots by six because the minimum work period length is three slots and the minimum rest period length is also three.

(3) *The Percentage of ATCo Working Time in Executive Positions Must Be between 40% and 60% of the Total Working Time (Not Including Rest Periods).* We divide the time that an ATCo is working in the executive position between the total working time in each row, yielding the percentage of working time in the executive position in that row ( $f_{1,3}$ ). If this is less than 40% or greater than 60%, we compute the difference to the nearest value (40 or 60 if lower or greater than the respective value), which we multiply by 0.4 (40%, the maximum difference) for normalization. Then, we add the differences for all rows in the solution matrix and divide by the number of rows:

$$g_{1,3} = \frac{1}{0.4} \sum_{i=0}^{n\text{ATCo}} \begin{cases} \frac{0.4 - f_{1,3}}{n\text{ATCo}}, & \text{if } f_{1,3} < 0.4 \\ \frac{f_{1,3} - 0.6}{n\text{ATCo}}, & \text{if } f_{1,3} > 0.6. \end{cases} \quad (15)$$

Since, as mentioned above, the three subobjectives are equally important, Objective 1 is

$$g_1 = \frac{g_{1,1}}{3} + \frac{g_{1,2}}{3} + \frac{g_{1,3}}{3}. \quad (16)$$

**3.3.2. Objective 2: Solution Structure Similar to Previous Template-Based Solution.** The second objective is to reach a solution structure that is similar to possible to the template-based solutions already used. This makes the solution easier to understand for the control center staff and would facilitate possible hand-made changes. To do this, the template-based structure was analyzed and used as a reference. The conclusion is that rest periods and work periods for the same sector should be as grouped as possible. The following function is used for this purpose:

$$f_2 = \sum_{i=1}^{n\text{ATCo}-1} \sum_{j=1}^{n\text{Slot}-1} \begin{cases} 1, & \text{if element}_{i,j} = \text{element}_{i+1,j} \\ 1, & \text{if element}_{i,j} = \text{element}_{i,j+1}. \end{cases} \quad (17)$$

This function runs through all the elements of the solution matrix checking if the right-hand ( $\text{sm}_{i,j+1}$ ) and bottom ( $\text{sm}_{i+1,j}$ ) elements contain the same sector and working position. If so, we add 1 to the function. Then, we perform the normalization:

$$g_2 = 1 - \frac{f_2^{\max} - f_2}{f_2^{\max}}, \quad (18)$$

where the theoretical maximum value is

$$f_2^{\max} = (n\text{Slot} - 1) \times (n\text{ATCo} - 1) \times 2, \quad (19)$$

derived without taking into account the constraints.

**3.3.3. Objective 3: Minimization of the Number of Control Center Changes.** This objective can be achieved by minimizing the number (not the duration) of rest periods. To do this, we count the number of res periods in each row. In order to normalize this value, the minimum value is matched to the number of rows,  $f_3^{\min} = n\text{ATCo}$ , since all ATCos must have at least one rest period, whereas the maximum value is computed as follows:

$$f_3^{\max} = \frac{n\text{Slot}}{6} \times n\text{ATCo}, \quad (20)$$

where the value 6 in the denominator denotes the sum of the minimum work and rest period lengths.

Finally,

$$g_3 = \frac{f_3^{\max} - f_3}{f_3^{\max} - f_3^{\min}}. \quad (21)$$

**3.3.4. Objective 4: Balancing ATCo Workloads.** The aim now is to balance the ATCo workloads to avoid high workloads for some ATCos and low workloads for the other. Thus, we will use the standard deviation of the work periods of the different rows in the solution matrix.

The maximum value of the standard deviation (regardless of the problem constraints) is equal to the mean. For example, if we have a workload of 200 slots, and we have four ATCos who can work a maximum of 100 slots, the mean will be 50 working slots. To maximize the variance, 100 working slots will be assigned to two ATCos and 0 to the other two ATCos. This would mean a variance equal to  $50^2$ ; that is, the standard deviation is 50, which matches up with the mean value. Then,

$$g_4 = \frac{\mu - f_4}{\mu}, \quad (22)$$

where  $\mu$  is the mean and  $f_4$  is the standard deviation of the ATCo workloads.

**3.4. Regular Expressions.** A *regular expression* (Regex or Regexp for short) is a special text string for describing a search pattern [23]. Regexs are composed of metacharacters (), [], and {}, characters classes like \A (start of string), \s (white space), or \d (digit), and quantifiers \* (0 or more), + (1 or more), and ? (0 or 1).

Their main application in programming is to automate search and analysis processes for use on many occasions. There are many tools whose pattern search engines use Regexs: Java (RegEx; see Package java.util.regex); JavaScript since version 1.2; Perl; C and C ++ (the library PCRE); PHP; Python; and .Net languages.

Regexs have a large number of applications, including pattern identification in DNA sequences [29], the reconstruction of DNA strands, and the derivation of a similarity measure for supervised learning [30] or in pattern recognition [31].

In our ATCo work-shift scheduling problem, Regexs are used to check the LC3, LC4, LC5, LC9, LC10, and LC11 working conditions. The verification of the other working conditions was programmed.

For instance, 12 regular expressions are used to check LC5 (ATCos cannot work for more than two consecutive hours) according to the following structure:

$$\begin{aligned} & \wedge (111) * [\text{a-zA-Z}] \{3,tMax\} (111) \\ & + [\text{a-zA-Z}] \{3,tMax\} (111) * \$ \end{aligned} \quad (23)$$

LC5 is verified if at least one out of the twelve Regexs is met. If we want to count the number of times LC5 is violated (part of the fitness function in the first phase), we go through the rows of the solution matrix checking the 12 Regexs and adding one to the penalty function when the corresponding row does not meet all 12 Regexs.

Regarding LC9 (the minimum work period length of an ATCo between two rest periods is 15 minutes), 12 Regexs are used again:

$$\begin{aligned} & \wedge (111) * [\text{a-zA-Z}] \{tMin,\} (111) \\ & + [\text{a-zA-Z}] \{tMin,\} (111) * \$ \end{aligned} \quad (24)$$

As in LC5, LC6 is verified if at least one out of the 12 Regexs is met, and we can go through the rows of the solution matrix



FIGURE 5: Solutions of the airspace sector configuration under consideration.

checking the 12 Regexs to count the number of times LC6 is violated.

Additionally, we can use Regexs to check if all the elements in the rows are 0 and, consequently, the respective ATCos are not working and can be removed: `Pattern.compile("^(111){"+nSlot+"}$$")`.

#### 4. Illustrative Example

We consider a control center in the Canary Islands (Spain) managing two cores. The approach sectors GCCCTM3 and

GCCCACC belong to Core 1, whereas sectors GCCCOCE, GCCCRE2, and GCCRU6 are en route sectors belonging to both cores (Core 1 and Core 2). Figure 2 shows the airspace sector configuration corresponding to a night shift for both cores.

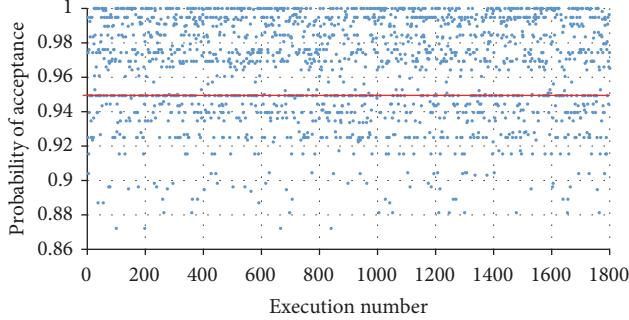
The number of available ATCos is 13, four of whom are CON ATCos and can work in the Core 1, and the other nine are PTD ATCos working in the Core 2.

First, we run the heuristic to build the initial solutions. Ten initial solutions are output using different templates, as described in Section 3.1. Figure 5(a) shows the solution built

TABLE 1: Comparing the objectives in Phase 2 between the initial feasible, the optimal, and the reference solutions.

	ATCo work and rest periods and positions			Control center changes Number of rest periods	ATCo workloads		
	Subobject. 1	Subobject. 2	Subobject. 3		$\sigma$	Min	Max
Initial sol.	1.5, 1.5, 20.7%	3.6, 21.8, 21.8%	7, 9, 10	56	87.38	120	390
Optimal sol.	34.7, 34.7, 68.4%	8.3, 41.6, 41.6%*	13, 13, 13*	35	30.91*	210	315
Reference sol.	20, 20, 100%*	0, 27.2, 27.2%	10, 11, 13	34*	58.42	180	345

\* denotes best values in the corresponding objectives.

FIGURE 6: Acceptance ratios for  $T_0 = 0.125$ .

using the template with a rest period lasting 12 slots (60 minutes). Note that as sector GCCCOCE is open throughout the night shift, it is covered by a special template with four ATCos (CON ATCo4, PTD ATCo1, PTD ATCo2, and PTD ATCo3) and rest periods of nine slots. The same applies for sectors GCCCACC and GCCCTM3, which can be viewed as a single sector open throughout the night shift and covered by the same special template with four ATCos (PTD ATCo4 to PTD ATCo7). Rest period lengths of 12 slots are considered for the other ATCos.

Note again that colors are used to represent sectors, whereas rest periods and out-of-shift periods are in white. The three-letter codes are used for sectors, and uppercase letters [A-Z] denote ATCos working as executives, whereas lowercase letters [a-z] denote planner positions.

The number of ATCos used in the initial solutions is always 14 but the labor conditions violated are different in each case. The only violated constraint in the solution illustrated in Figure 5(a) is the number of ATCos used (14) which is greater than the available number of ATCos (13). All working conditions are met.

Then, the first MIR of simulated annealing is conducted with the following parameter values: we run the algorithm proposed in [26] with different initial temperatures for adjustment using an acceptance ratio of 0.95, leading to  $T_0 = 0.125$ ; see Figure 6. The number of iterations during which the temperature does not change is  $L = 1200$  and, following [27], we set  $\alpha = 0.9$ . The search stops when the fitness of the best solution does not improve by at least 0.005% during 30000 iterations.

The MIR of simulated annealing yielded different feasible solutions with 14 ATCos. Figure 5(b) shows the feasible solution which is reached when simulated annealing starts

the search from the initial infeasible solution shown in Figure 5(a).

Figure 7(a) shows the evolution of the fitness value during the search process, clearly illustrating the iteration where the ATCo number decreases in the first iterations of the algorithm and how the number of violated working conditions decreases throughout the iterations until a feasible solution is reached (fitness value = 1).

A second MIR of simulated annealing is executed, in order to output optimal solutions starting from the different feasible solutions derived in the previous phase. Now,  $T_0 = 0.075$ ,  $L = 3000$ ,  $\alpha = 0.9$ , and the search stops when the fitness of the best solution does not improve by at least 0.001% during 50000 iterations.

Figure 5(c) shows the optimal solution, which is reached when simulated annealing starts the search from the initial feasible solution shown in Figure 5(b).

Figure 7(b) shows the evolution of the optimal value of the fitness value during the search process and the values for the four objectives under consideration.

Figure 5(d) shows the reference solution yielded by the tools available before the proposed template-based approach. First, the optimal solution which is reached using the solution method proposed in this paper outperforms the reference solution; the fitness values for both solutions are 0.8860 and 0.8838, respectively.

Note that the structure of the optimized solution (see Figure 5(c)) is more similar to the reference solution than the initial feasible solution (see Figure 5(b)). This was the second most important objective in the second phase of the problem-solving method. Moreover, Table 1 shows information regarding the other objectives considered in the second phase for the feasible initial, optimal, and reference solutions.

The first three columns in Table 1 refer to the targets fixed for the ATCo work and rest periods and positions (Objective 1): 45 minutes for the time ATCos remains in the same sector and working position, 90 minutes for the working time between rest periods and a ratio of from 40% to 60% for ATCo working times in executive positions to total working time. In subobjetive 1, the three values correspond with the percentage of work periods in which the time ATCos remain in the same sector and working position differ in less than 10, 15, and 25 minutes from 45, respectively. In subobjetive 2, the three values correspond with the percentage of working times between breaks differing in less than 15, 20, and 25 minutes from 90, respectively. Finally, in the third subobjetive, the three values correspond with the number of ATCos that

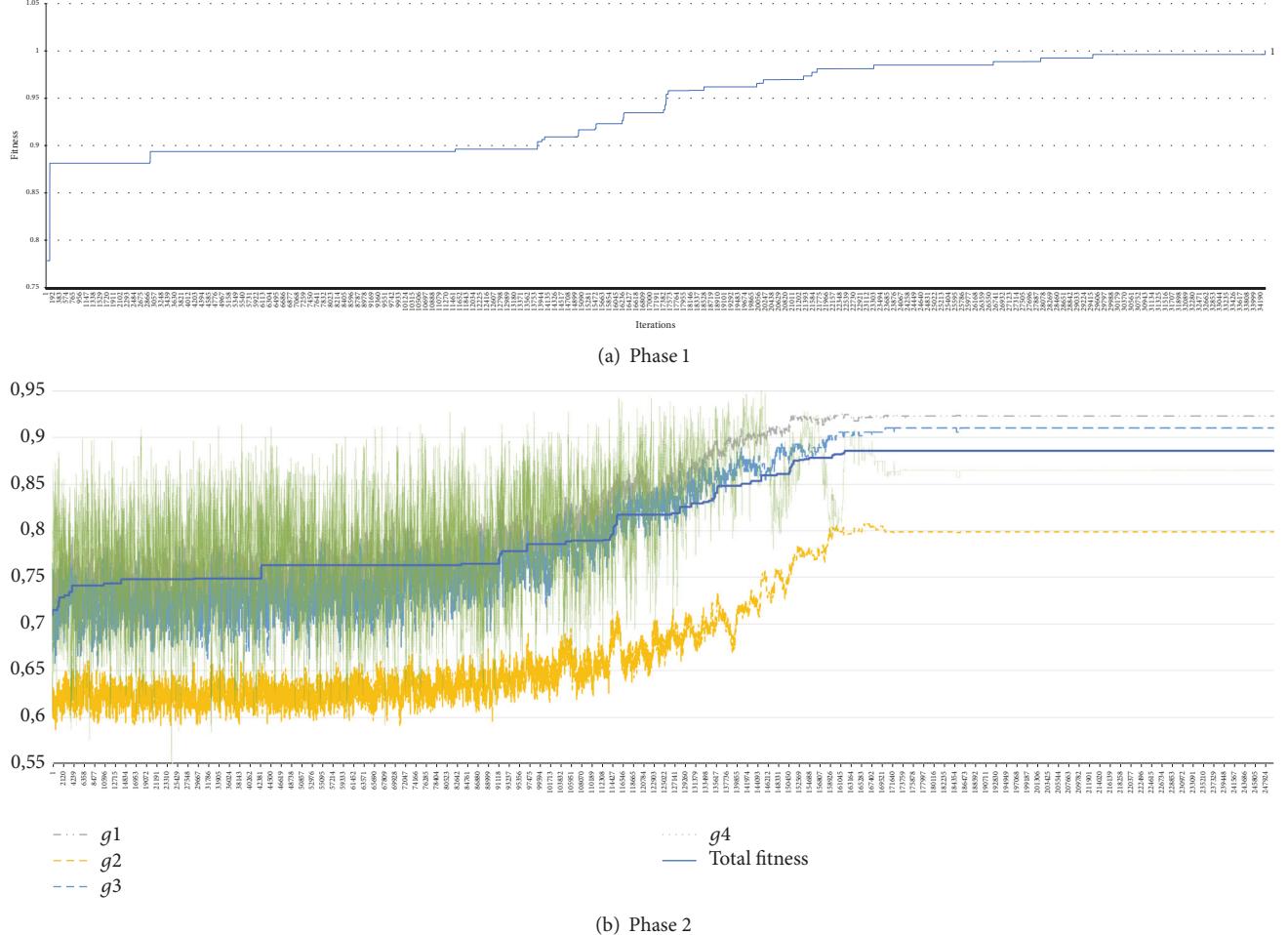


FIGURE 7: Fitness evolution in Phases 1 and 2.

verify the condition, those differing in less than 5, 10, and 15% from the target, respectively.

We can find that the optimal solution outperforms the reference one in subobjectives 2 and 3. Moreover, taking into account the fact that the three subobjectives are equally important, the reference solution (0.9274) slightly outperforms the optimal one (0.9218).

The number of control center (CC) changes (Objective 3) was minimized by counting the number of rest periods. Its value in our optimal solution is 35; it is 56 and 34 for the initial and reference solutions, respectively.

Finally, regarding the homogeneous distribution of ATCo workloads (Objective 4), the ATCo workload standard deviation for our optimal solution is the lowest (30.91). Moreover, ATCo workloads are within the interval [210, 315] in the optimal solution, which is narrower than [180, 345] corresponding to the reference solution.

We can conclude that our optimal solution is very similar to the reference one regarding the target fixed for the ATCo rest periods and positions and the number of control center changes, and the ATCo workload is more balanced than for the reference solution.

## 5. Conclusions

We proposed a new approach to solve the ATCo work-shift scheduling problem that, given a fixed number of ATCos and an airspace sector configuration to cover, optimizes several objectives dealing with the ATCo work and rest periods and positions, the structure of the solution, the number of control center changes, or the distribution of the ATCo workloads, while satisfying a set of ATCo working conditions according to Spanish regulations.

This is a three-phase approach: a heuristic to build initial infeasible solutions and two multiple independent runs of the simulated annealing metaheuristic. The first run aims to achieve feasible solutions, whereas the second, where only feasible solutions are visited, aims to achieve an optimal solution transforming the multiobjective into a single optimization problem by means of the rank-order centroid function, which exploits ordinal information about the relative importance of the objectives provided by the experts.

Regular expressions are used to check working conditions in the solutions, thus providing high testing speed and

modularity for a clear and maintainable implementation of the optimization model.

We illustrated the proposed approach using a real example and compared its performance with a template-based reference solution. Note that our optimal solution and the benchmark solution have a similar structure, which was one of our stated goals. Their performance is very similar regarding the target fixed for the ATCo rest periods and positions and the number of control center changes. However, our optimal solution has a more balanced ATCo workload than the reference solution.

The developed software is currently being used to solve more complex airspace sector configurations at Barcelona-El Prat control center, outperforming the reference solutions in all cases.

One line of future work that we propose is to use additional regular expressions to verify other working conditions, which are now programmed. Besides, we have to test more airspace sector configurations, some of which should represent extreme cases, in order to ensure proper operation before its adoption at control centers.

Besides, we have considered the use of ordinal information and the ROC method as a starting point to represent the relative importance of objectives. This is less demanding for experts. However, a sensitivity analysis is currently being carried out with the support of experts on the basis of different instances of the problem to adjust these weights. For instance, tradeoffs between the performance with respect to the target fixed for the ATCo rest periods and positions and the structural similarity between our optimal solution and the reference solution have to be considered.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The research reported in this paper was supported by Spanish Ministry of Economy and Competitiveness Projects MTM2014-56949-C3-2-R and MTM2017-86875-C3-3R. The authors are also grateful to Enrique Pintos, Miguel García, and Guillermo Orenga, from the Reference Center for Research, Development and Innovation in ATM, whose expertise enhanced the conducted research.

## References

- [1] J. Telhada, “Alternative MIP formulations for an integrated shift scheduling and task assignment problem,” *Discrete Applied Mathematics*, vol. 164, no. 1, pp. 328–343, 2014.
- [2] G. Post, L. Di Gaspero, J. H. Kingston, B. McCollum, and A. Schaefer, “The Third International TIMetabling Competition,” *Annals of Operations Research*, vol. 239, no. 1, pp. 69–75, 2016.
- [3] G. Post, J. H. Kingston, S. Ahmadi et al., “XHSTT: an XML archive for high school timetabling problems in different countries,” *Annals of Operations Research*, vol. 218, no. 1, pp. 295–301, 2014.
- [4] G. Fonseca, H. Santos, T. Toffolo, S. Brito, and M. Souza, “GOAL solver: a hybrid local search based solver for high school timetabling,” *Annals Operations Research*, vol. 239, no. 1, pp. 1–21, 2014.
- [5] A. Kheiri, E. Ozcan, and A. J. Parkes, “HySTT: Hyper-heuristic search strategies and timetabling,” in *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT ’12)*, pp. 497–499, 2012.
- [6] J. Romros and J. Homberger, “An evolutionary algorithm for high school timetabling,” in *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT ’12)*, pp. 485–488, 2012.
- [7] M. Sorensen, S. Kristiansen, and T. Stidsen, “International timetabling competition 2011: an adaptive large neighborhood search algorithm,” in *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT ’12)*, pp. 489–492, 2012.
- [8] G. H. Fonseca and H. G. Santos, “Variable neighborhood search based algorithms for high school timetabling,” *Computers & Operations Research*, vol. 52, no. part B, pp. 203–208, 2014.
- [9] O. G. Czibula, H. Gu, and Y. Zinder, “Timetabling of workplace training: a combination of mathematical programming and simulated annealing,” in *Proceedings of the 11th International conference on Practice and Theory of Automated Timetabling (PATAT ’16)*, pp. 95–107, 2016.
- [10] L. N. Ahmed, E. Özcan, and A. Kheiri, “Solving high school timetabling problems worldwide using selection hyper-heuristics,” *Expert Systems with Applications*, vol. 42, no. 13, pp. 5463–5471, 2015.
- [11] G. H. Fonseca, H. G. Santos, and E. G. Carrano, “Integrating matheuristics and metaheuristics for timetabling,” *Computers & Operations Research*, vol. 74, pp. 108–117, 2016.
- [12] H. Babaei, J. Karimpour, and A. Hadidi, “A survey of approaches for university course timetabling problem,” *Computers & Industrial Engineering*, vol. 86, pp. 43–59, 2015.
- [13] I. A. Chaudhry and A. A. Khan, “A research survey: review of flexible job shop scheduling techniques,” *International Transactions in Operational Research*, vol. 23, no. 3, pp. 551–591, 2016.
- [14] M. A. Cruz-Chávez, M. Martínez-Rangel, and M. Cruz-Rosales, “Accelerated simulated annealing algorithm applied to the flexible job shop scheduling problem,” *International Transactions in Operational Research*, vol. 24, no. 5, pp. 1–19, 2017.
- [15] T. M. Días, D. F. Ferber, C. C. de Souza, and A. V. Moura, “Constructing nurse schedules at large hospitals,” *International Transactions in Operational Research*, vol. 10, no. 3, pp. 245–265, 2003.
- [16] C. C. Ribeiro, “Sports scheduling: problems and applications,” *International Transactions in Operational Research*, vol. 19, no. 1–2, pp. 201–226, 2012.
- [17] M. Arnving, B. Beermann, B. Koper et al., “Managing shiftwork in European ATM, literature review,” *European Organization for the Safety of Air Navigation*, 2006.
- [18] EUROCONTROL, “Shiftwork practices study - ATM and related industries, DAP/SAF-2006/56 Brussels: EUROCONTROL, 2006”.
- [19] “EATCoHIP Human Resources Team (1998). ATM manpower planning in practice: Introduction to a qualitative and quantitative staffing methodology. HUM.ET1.ST02.2000-REP-01 Brussels: EUROCONTROL”.
- [20] M. Stojadinovic, “Air traffic controller shift scheduling by reduction to CSP, SAT and SAT-related Problems,” in *Proceedings of*

*the 20th International Conference on Principles and Practice of Constraint Programming (CP '14), pp. 886–902.*

- [21] A. Biere, M. Heule, H. van Maaren, and T. Walsh, “Handbook of satisfiability,” in *Frontiers in Artificial Intelligence and Applications*, vol. 185, IOS Press, 2009.
- [22] M. Stojadinovic, “Hybrid of hill climbing and SAT solving for air traffic controller shift scheduling,” *Journal of Information Technology and Applications*, vol. 2, pp. 81–87, 2016.
- [23] J. E. F. Friedl, “Mastering Regular Expressions,” *O'Reilly and Associates*, 1997.
- [24] V. Cerny, “Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm,” *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
- [25] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *American Association for the Advancement of Science: Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [26] W. Ben-Ameur, “Computing the initial temperature of simulated annealing,” *Computational optimization and applications*, vol. 29, no. 3, pp. 369–385, 2004.
- [27] B. Hajek, “Cooling schedules for optimal annealing,” *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.
- [28] J. Butler and D. L. Olson, “Comparison of centroid and simulation approaches for selection sensitivity analysis,” *Journal of Multi-Criteria Decision Analysis*, vol. 8, no. 3, pp. 146–161.
- [29] F. Marass and C. Upton, “Sequence Searcher: A Java tool to perform regular expression and fuzzy searches of multiple DNA and protein sequences,” *BMC Research Notes*, vol. 2, article 14, 2009.
- [30] I. Spasić, G. Nenadić, K. Manios, and S. Ananiadou, “Supervised Learning of Term Similarities,” in *Intelligent Data Engineering and Automated Learning (IDEAL)*, pp. 429–434, Springer, Berlin, Germany, 2002.
- [31] D. Duc An Bui and Q. Zeng-Treitler, “Learning regular expressions for clinical text classification,” *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 850–857, 2014.

