Abstract—In this paper we propose a new approach to solving the air traffic controller (ATC) work shift scheduling problem. This approach that minimizes the number of ATCs required to cover a given airspace sectoring, while satisfying a set of ATC labor conditions. First, initial feasible solutions are built using a heuristic, and then multistart simulated annealing is used to reach optimal solutions. In the search process, we use regular expressions to check the feasibility of the visited solutions. This provides high testing speed. Once the optimal ATC number is reached, it is used as the initial solution for a new optimization process aimed at balancing the ATC workloads.

I. INTRODUCTION

The core of air traffic controller (ATC) activity is to facilitate the airspace and airport surface traffic flow under their responsibility, while avoiding collisions between aircrafts. To satisfy this essential safety constraint, they must detect and solve possible conflicts between trajectories.

The airspace is composed of volumes (elementary unit of airspace). As a human ATC 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 and are operated by two ATC working positions (executive and planner). All the sectors open at any one time must cover all the volumes of the corresponding airspace. This is referred to as airspace sectoring. The sectorization changes throughout the day depending on the aircraft traffic. A higher volume of air traffic involves opening more sectors and, consequently, more ATCs are necessary.

The sectorization required to handle the traffic estimations for a time period can be designed beforehand. Therefore, a very important problem in air traffic control is to determine the minimum number of ATCs necessary to cover a sectorization structure for a given time period, denoted as airspace sectoring, while satisfying certain strong constraints accounting for ATC labor conditions, including, for instance, resting and working time distributions.

This optimization problem belongs to the class of timetabling and scheduling problems. Timetabling problems require the assignment of times and resources to events, taking into account sets of required and desirable constraints. 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 [15],[8]. In air traffic management (ATM) context the number of publications is increasing. Some of the solved problems, such as Gate Assignment Problem or Aircraft Sequencing and Scheduling Problem, could be formulated as Job Shop Scheduling Problems. Many of these problems are solved by metaheuristics and published in different papers. A review and classification of some of them is in [12].

Regarding the scheduling problems in the context of ATM, an overview of available results related to ATC shift scheduling is presented in [1]. There are already some software tools for generating ATC schedules [6], and their advantages and drawbacks have already been recognized [5]. Some are in-house tools, and details are not available for most of them. Stojarjovic [13] describes in depth three encodings of the problem and proposes three optimization techniques for solving the problem. Later, Stojarjovic [14] presented a new hybrid technique for solving the ATC scheduling problem, which is a combination of propositional satisfactory problem solving [3] and hill climbing. Tello et al. [7] propose a similar methodology to solve ATC Work Shift Scheduling Problem using different objectives and constraints.

In this paper we deal with an ATC work shift scheduling problem for covering control positions (sectors), aimed at minimizing the number of ATCs needed for a given airspace sectoring, while satisfying some strong constraints accounting for ATC labor conditions. To do this, we propose a novel methodology, wherein a heuristic is used to build initial feasible solutions and then multistart simulated annealing (SA) is used to reach an optimal solution.

Regular expressions (Regexp) [9] are used to check the feasibility of the visited solutions in the search process. A Regexp is a sequence of characters that define a search pattern, which is matched to the string representation of every solution. In our context, the patterns represent the violation of ATC labor conditions. The benefits of using RegExp 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 ATC work shift scheduling problem, together with the constraints accounting for ATC labor conditions. The proposed problem-solving methodology is described in Section 3. First, we introduce a heuristic to derive initial feasible solutions in Section 3.1. Next, regular expressions and their application for checking the feasibility of candidate solutions are introduced in Section 3.2. Finally, the adaptation of simulated annealing to derive optimal solutions is described in Section 3.3. An example is used to illustrate the proposed methodology and how the optimal solution reached...
outperforms the reference solution by one ATC in Section 4. Finally, some conclusions are provided in Section 5.

II. PROBLEM DESCRIPTION

One of the core tasks of ATCs is to avoid collisions between aircraft. To do this, the ATC must assure that the aircraft are always separated by a minimum safety distance, denoted separation standard. To do this, air traffic controllers issue instructions to pilots to maneuver the aircraft laterally, vertically, or by adjusting speed or rate of climb/descent.

ATCs can perform two functions: executive and planner. The executive ATC talks to the aircraft and gives instructions to the pilots to avoid conflict situations between aircraft, whereas the planner ATC is responsible for anticipating possible conflicts between aircraft and communicating with the executive ATC to solve the situation before it happens.

As already mentioned, as an ATC can only handle a limited amount of traffic, the airspace is divided into a number of sectors, each operated by an executive and a planner ATC.

All the sectors open at any one time must cover all the volumes of the corresponding airspace. This is referred to as a airspace sectoring which changes throughout the day depending on the air traffic. A higher volume of air traffic means more sectors with smaller dimensions will be open, thus requiring more ATCs. As a result, sectors are dynamically divided and merged over time depending on the air traffic, and the number of ATCs necessary to cover the open sectors varies accordingly. The airspace sectoring needed to handle the traffic estimations for a time period (usually a day) can be designed beforehand. Thus, the airspace sectoring contains the sectors open over a twenty-four hours period.

Figure 1 shows an example of an airspace sectoring for Madrid Path1 in Spain. Each interval is associated with a configuration (1A, 2A, 4A, . . . ), where the integer value represents the number of open sectors and the letter refers to the sector configuration, i.e. there are two sectorizations with the same number of sectors but different spatial distributions (6B and 6C in Figure 1).

![Fig. 1. Madrid Path1 airspace sectoring](image)

In Figure 1, sectorization 1A is used during the first four and a half hours, all airspace is covered by a sector. Then, the sector is divided into two new sectors leading to sectorization 2A, which is established for one hour. Next, sectorization 4A is used during an hour and so on. Note that the 24-hour period has been divided into night (N), morning (M) and afternoon (A) periods in Figure 1.

Besides, five different ATC shifts are considered (see Figure 2): long morning (LMS) (5:40-14:00), long afternoon (LAS) (14:00-22:20) and night (NS) (21:20-6:40), for ATCs working in NS at night and LMS shifts in the time period (5:40-6:20), and for ATCs working in MS and LMS shifts in the morning. In the time period (22:20-5:40), only the NS ATCs are working.

![Fig. 2. Daily working shifts](image)

Besides, a number of constraints accounting for ATC labour 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.

ATC labor conditions are as follows:

1. ATCs can cover a position (as executive or planner) in only one sector at any time.
2. Sectors must be covered by two ATCs (an executive and a planner ATC) during the time they are open.
3. ATCs must rest during 25% of the working shift in day shifts (MS, LMS, AS and LAS). A 33% resting time is established for the night shift (NS).
4. ATCs cannot work for more than two consecutive hours.
5. Each rest period should last at least half an hour.
6. ATCs must remain in the same sector and position for at least half an hour (minimum time in a position).
7. Each ATC can work at most in two different sectors in the respective shift.
8. ATCs should occupy the planner (executive) position approximately 60% (40%) of the time.
9. ATCs cannot work for more than eight hours a day.
10. ATCs are assigned to one shift and cannot work in any other shift.
11. A sector change is not allowed without resting unless the sectorization is modified at this time. ATCs can change to an affine sector (two sectors are affine if they have at least one common volume) without resting, but are required to work in the new sector for at least 15 minutes. Otherwise, ATCs need to rest to perform the sector change.

The objective of the problem is to determine the minimum number of ATCs necessary to cover a given airspace sectoring while satisfying the above ATC labor conditions. Moreover, it would be interesting to balance the ATC workload, avoiding big differences among ATCs.
III. PROBLEM-SOLVING METHODOLOGY

In this section, we introduce a methodology to solve the ATC work shift scheduling problem under consideration for a given airspace sectoring. The proposed methodology is based on three elements, a heuristic to build initial feasible solutions, the use of regular expressions (Regex or Regexp) to check quickly the feasibility of the solutions according to the Spanish normative and a multistart simulated annealing (SA) to achieve optimal solutions.

Before describing the above elements in detail, let us first examine how the proposed methodology represents the solutions. We use a matrix containing 84 columns, i.e., the number of time slots of five minutes under consideration, and each row is associated with an ATC. A five minute slot is considered because it is the greatest common divisor between the times of constraints (e.g., ATCs are required to work in the new sector for at least 15 minutes) and the minimum time that a sector can be open (20 minutes). The number of rows is established when applying the heuristic to build an initial solution. Note that initial solutions may have different numbers of ATCs.

Each element of the matrix \((i, j)\) represents the state of the ATC \(i\) in the time slot \(j\). We use the value 1 to represent a resting ATC, uppercase letters [A-Z] to point out that the ATC is working as an executive ATC in the corresponding sector, whereas lowercase letters [a-z] are used for planner positions. Finally, value 0 represents that the ATC is out of shift, i.e., neither working nor resting.

Colors are used to represent sectors when displaying solutions, see Figure 3. Resting periods and out of shift periods are in blue and white, respectively. The airspace sectoring in Figure 3 (11A, 13F, 11A) accounts for six hours (5:20 - 11:20) rather than the whole day, and it is covered by 39 ATCs (numbers of rows).

![Fig. 3. Solution representation](image3.png)

III-A. A heuristic for the construction of initial feasible solutions

We propose a heuristic to build a set of initial solutions with different resting periods. They will be used afterwards in the multistart simulated annealing to reach an optimum solution. The proposed heuristic is based on the use of an optimized template, see Figure 4, with three ATCs covering a sector during 96 time slots (eight hours) without specifying the ATC positions (executive or planner), which are assigned afterwards.

![Fig. 4. Template](image4.png)

In the template, the duration of the working periods is always the same and is twice as long as the resting periods. Moreover, if different resting period durations are used, the heuristic will build different initial solutions, albeit with a similar structure. As the labor conditions establish that each resting period should last at least half an hour (six slots), and the minimum and maximum working periods are six and twenty four slots, respectively, the duration of resting period must be between six and twelve slots. Thus, the heuristic build seven different initial solutions.

The steps of the heuristic are as follows:

- **Step 1. Add templates to cover the airspace sectoring.** For each time slot \(l = 0, \ldots, 288\) do
  - Identify the number of open sectors in slot \(l, s_l\).
  - For each open sector \(k_l\) in time slot \(l\), \((k_l = 1, \ldots, s_l)\) do
    - If no ATC is available in any added template for slot \(l\), then we add a new template and assign two ATCs to sector \(k_l\) in the new template without establishing their positions.
    - If ATCs are available (in one or more templates), we check if sector \(k_l\) was assigned to two ATCs who are available in slot \(l\). In that case, we assign sector \(k_l\) to these ATCs in order to avoid unnecessary sector changes. Otherwise, sector \(k_l\) is assigned to two ATCs who were resting in slot \(l - 1\) and are available in slot \(l\). If this is not possible, we add a new template.
    - Working periods not assigned to any sector are transformed into resting periods.

- **Step 2. Allocation of positions (executive or planner).** For each time slot \(l = 0, \ldots, 288\) do
  - Identify the number of open sectors in slot \(l, s_l\).
  - For each open sector \(k_l\) in time slot \(l\), \((k_l = 1, \ldots, s_l)\), ATCs assigned to the sector are identified.
    - If both ATCs were assigned to that sector in the previous slot, they continue in the same position.
    - If one ATC was assigned to that sector in the previous slot, s/he continues in the same position and the other ATC takes up the non-covered position.
    - Otherwise (both ATCs were resting in slot \(l - 1\)), ATC positions are assigned at random.
• Step 3. Improving feasibility. For each ATC, if the minimum working period duration is not satisfied, we try to reallocate that working period to another ATC. To do this:
  • If another ATC working in the same sector and position, started resting at that time and the resting period was greater than the working period for reallocation, then the working period is assigned to that ATC.
  • If another ATC begins working in that sector and position when the working period for reallocation ends, and the resting period that ends is greater than the working period that we are trying to reallocate, then the working period is assigned to that ATC.
• Step 4. Check the feasibility. Regular expressions are used to check the labour conditions on the built solution.

Note that some of the solutions output by the heuristic may not be feasible and, consequently, they are discarded as starting solutions for simulated annealing.

III-B. Regular expressions

A regular expression (Regex or Regexp for short) is a special text string for describing a search pattern [9]. RegExs are composed of metacharacters (), [], { }, [a-zA-Z], and quantifiers * (0 or more), + (1 or more) and ? (0 or 1). For instance, the pattern “car(s)\+” matches the words “car” and “cars”, and \( ^{\w+[@a-zA-Z.\-\_0-9]+[a-zA-Z.\-\_0-9][2,6]} \) could be used to identify email addresses in a text, like U123456789@gmail.com.

The benefits of using RegExps are high testing speed and modularity for a clear and maintainable implementation of the optimization model.

In our ATC work shift scheduling problem, RegExs are used to check the feasibility of the solutions, i.e. the constraints representing the ATC labor conditions. The solution codification described at the beginning of Section 3 has to be taken into account to build RegExs. For example, the labor condition that ATCs cannot work more than two consecutive hours can be verified using eight patterns. Each pattern accounts for a possible number of resting periods for any ATC. It is necessary to match only one of the eight patterns to satisfy the constraint.

The pattern accounting for no resting periods is
\[ ^{0*\{[a-zA-Z]\{1,24\}\}} \] 0*\$, which accepts strings with 1 to 24 (lower and uppercase) letters (5 minutes 24 slots = 2 hours).

The following pattern accounts for one resting period:
\[ ^{0*\{[a-zA-Z]\{1,24\}\}1\{6\}[,a-zA-Z]\{1,24\}\}0*\$, for two resting periods:
\[ ^{0*\{[a-zA-Z]\{1,24\}\}1\{6\}[,a-zA-Z]\{1,24\}\}1\{6\}[,a-zA-Z]\{1,24\}\}0*\$, and so on up to seven resting periods.

RegExs are also used to check that ATCs work in only one shift, ATCs do not work more than 24 consecutive slots (two hours), ATCs work at least six consecutive slots in the same position, no position change is performed without resting, and all opened sectors are covered by two ATCs (executive and planner).

Other labor conditions, including constraints regarding the total resting time, affine sectors treatment and the maximum number of work positions for each ATC are verified using ad hoc java code instead RegEx based patterns.

III-C. Simulated Annealing

Simulated annealing (SA) [11], [4] 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 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.

The initial solution will be one of the feasible solutions output by the heuristic described in Section 3.1. However, the initial solution is reorganized before starting the search. Specifically, the ATC with the least workload is placed on the first row of the solution codification, the ATC with the second least workload is placed on the second row, and so on, where the ATCs with the highest workloads are placed on the last rows. As pointed out in the SA algorithm, the above reorganization is performed on each of the visited solutions in the search process.

A common cooling schedule is to keep the temperature value constant for a number of iterations \( (L) \) and then decrease it using a geometric schedule: \( T_k = \alpha^k T_0 \), the typical value for \( \alpha \) being 0.95 [10].

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

1. An ATC is selected at random. The aim is to reallocate some of this ATC’s workload to another ATC.
2. A working period of the chosen ATC is selected at random. If it is composed of more than 12 slots, only the first 12 slots will be considered for reallocation, otherwise the whole working period is used.
3. We randomly select a second ATC and check if the whole working period for reallocation can be assigned to that ATC. If this is not possible, then we try with another ATC and so on. To do this, we check that the ATC under consideration is resting in the slots corresponding to the working period to be reallocated. If it is not possible to reallocate the considered working
period to any ATC, then we reduce it by one slot (the last one) and repeat the process. We repeat the process until the working period is reallocated or is reduced to two slots, in which case, we go back to Step 1.

Finally, regarding the objective function, the goal is to minimize the number of ATCs needed to cover all open sectors in a given airspace sectoring. The objective function that we consider in the search process is:

\[ \max z = \frac{1}{c^2} \sum_{i=1}^{c} h_i, \]

where \( c \) is the number of ATCs and \( h_i \) is the number of slots that the \( i \)-th ATC is working in. The term \( \frac{1}{c^2} \) implies that when the number of ATCs decreases the objective value is greatly improved. For a given number of ATCs, objective values are higher for solutions with a high workload for ATCs with higher indexes (in the last rows of the solution codification).

Thus, if we take into account the considered objective function and the row reorganization of the visited solutions, we find that the search process tends to reallocate working periods from ATCs in the first rows to ATCs in the last rows (with the highest workload). In this way, we increase the resting periods among the first ATCs and tend to decrease the number of ATCs needed to cover the airspace sectoring.

However, once the minimum number of ATCs is reached, the resulting solution is not balanced in the sense that the workloads for ATCs in the last rows are very high, whereas workloads for ATCs in the first rows are low.

To overcome this drawback, we propose using the solution reached as the initial solution of a new optimization problem. The objective function of this optimization problem aims to balance workloads for a fixed number of ATCs (output by the optimal solution), represented by the standard deviation of the ATC workloads, wherein the ATC selection in the neighborhood structure is completely at random.

IV. ILLUSTRATIVE EXAMPLE

We consider the airspace sectoring shown in Figure 5 to illustrate the proposed methodology. There are 11 open sectors from 5:20 to 8:20, 13 open sectors from 8:20 to 10:40 and 11 from 10:40 to 11:20.

![Fig. 5. Airspace sectoring](image)

First, we run the heuristic to build the initial feasible solutions. Seven initial feasible solutions are obtained using different templates, as described in Section 3.1. Figure 3 shows the solution built using the template with a resting period lasting 10 slots (50 minutes). Note again that colors are used to represent sectors, whereas resting periods and out of shift periods are in blue and white, respectively. Uppercase letters [A-Z] denote ATCs working as executives, whereas lowercase letters [a-z] denote planner positions.

Then, multistart simulated annealing (SA) is carried out with the following parameter values: we run the algorithm proposed in [2] with different initial temperatures for adjustment using an acceptance ratio 0.9, leading to \( T_0 = 0.75 \). The number of iterations during which the temperature does not change is \( L = 500 \) and, following [10], we established \( \alpha = 0.9 \). The search stops when the objective of the best solution does not improve by at least 0.05% during 1750 iterations.

The multistart SA reached seven solutions with the following number of ATCs \{35, 34, 34, 35, 34, 34, 34\}. To do this, we used a PC Intel i5-3230M CPU on 2.60GHz with 8GB of RAM running on Windows 10. It took 0.17 minutes to build the initial solutions and 3.5, 4, 9.6, 4.3, 13.2, 3.7 and 4.4 minutes, respectively, to reach the seven optimal solutions. Figure 6 shows the best solution, with 34 ATCs, derived from the initial feasible solution shown in Figure 3.

![Fig. 6. Solution achieved with our methodology](image)

Figure 7 shows the reference solution including 35 ATCs. This is the solution yielded by the tools available before the proposed approach. The average ATC workload and the standard deviation are 280 minutes and 43.72 for the reference solution, respectively. Thus, the solution archived outperforms the benchmark since 34 ATCs are necessary to cover the airspace sectoring, with an average ATC workload of 288.23 minutes and a standard deviation 64.28. Looking at the standard deviation, however, we realized that the ATC workload is more balanced in the reference solution.

Using the solution achieve (Figure 6) as the initial solution in the new optimization problem aimed at balancing the 34 ATC workloads, we obtain the solution shown in Figure 8, with the same average workload but a lower standard deviation, 16.17.

To conclude, the proposed methodology clearly outperforms the reference solution since one less ATC is needed to cover the airspace sectoring, and the ATC workloads are more balanced.

V. CONCLUSIONS

We have proposed a new approach to solving the ATC work shift scheduling problem that minimizes the number
of ATCs necessary to cover a given airspace sectoring while satisfying a set of ATC labour conditions according to Spanish regulations. The approach consists of three elements: a heuristic to build initial feasible solutions, the use of regular expressions to check the feasibility of the solutions, thus providing high testing speed and modularity for a clear and maintainable implementation of the optimization model, and multistart simulated annealing to reach good solutions.

Once the ATC number can not be reduced (optimal or close to the optimum), it is used as the initial solution of a new optimization process aimed at balancing the ATC workloads.

The proposed approach has been illustrated using a real example of Spain, and the optimal solution reached outperforms the reference solution by one ATC and with more balanced workloads.

ACKNOWLEDGMENT

The research reported in this paper was supported by Spanish Ministry of Economy and Competitiveness projects MTM2014-56949-C3-2-R and MTM2017-83132-C2-2-R.

REFERENCES