AIRLINE CREW SCHEDULING IN THE BRAZILIAN CONTEXT

Wagner de Paula Gomes (wagner.gomes@poli.usp.br)

Nicolau D. Fares Gualda (ngualda@usp.br)

Department of Transportation Engineering,

Escola Politécnica, University of São Paulo

ABSTRACT

This paper treats the Crew Scheduling Problem (CSP), important part of the airlines operational planning. The CSP is usually divided, in the literature, into two subproblems, formulated and solved sequentially: Crew Pairing Problem (CPP) and Crew Rostering Problem (CRP). This decomposition is justified by its combinatorial nature, but it does not provide a global treatment to the CSP, in terms of cost and quality of final solution. Therefore, the state of the art involves the integrated solution of CSP, with both subproblems (CPP and CRP) solved simultaneously. The problem, however, is NP-Hard. The methodology proposed in this paper aims to obtain an integrated solution of the CSP through a hybrid genetic algorithm associated with a depth-first search procedure, taking into account the proper crew legislation. The methodology was tested, with success, to solve instances related a network of a Brazilian airline.

Keywords: air transportation, airline crew scheduling, metaheuristic, hybrid genetic algorithm, depth-first search.

1. INTRODUCTION

After the fuel costs, crew costs constitute the second largest expenses of an airline (about 20% of the total operational cost). Thus, efficient crew planning is of major importance for airlines.

The main objective of the Crew Scheduling Problem (CSP) is to assign a set of flights planned for a given period of time to a set of crew members, considering the labor regulations, safety rules and policies of airlines, such that the crew total cost is minimal.

The CSP is NP-Hard (Andersson et al., 1998), given the large amount of rules and regulations, making it difficult (or even impossible), in real world instances, to get its solution by exact methods, leading to the need to use heuristics or metaheuristics methods.

The methodology proposed in this paper aims to obtain an integrated modeling of the CSP through a hybrid genetic algorithm associated with a depth-first search procedure, taking into account the proper crew legislation.

This paper is organized as follows: in Section 2 an overview of the CSP and of the problem in the literature is presented. Section 3 describes the proposed methodology and Section 4 presents the results of tests and practical applications of the model. Section 5 presents conclusions of the research.

2. DESCRIPTION OF THE PROBLEM

The coverage of each flight planned by the airline requires a set of crew members of distinct categories: technical and non-technical ones. The technical crew members (captain and first officer) are qualified to fly a specific aircraft type or a set of closely related aircraft types, known as a *fleet or aircraft family*. The non-technical crew members (flight attendants) may be qualified to fly a larger set of different aircraft types. Thus, the CSP can be solved separately for each crew category (technical or non-technical), without loss of quality, considering only the flights assigned to a particular aircraft type.

The CSP treated in this paper is defined as the problem of assigning a set of flights of a given aircraft type to a set of crew members of the same category (in this case, only technical crew) qualified to fly one type of aircraft. Each crew member has an individual calendar of availability, which takes into account a set of previously assigned activities, such as training periods, vacation, medical exams, days off, alert duties and reserve duties. In addition, each crew member is stationed on a crew base (or home base), that is the location where he receives his days off.

The input of the CSP is the set of flights to be covered. Initially, the flights are grouped to form duty periods, that are series of sequential flights comprising a day’s work for a crew member. Then, the duty periods are assigned to the crew members, considering the rules and regulations, the crew members’ availabilities, the coverage of all flights exactly once, and the minimization of crew total cost.

The rules and regulations applicable to CSP in the Brazilian context (that complies with international ones) are (ANAC, 2009; SNA, 2009):

1. On a duty period, the flights must be sequential in time and space;

2. The interval between two consecutive flights must be within a minimum and maximum time, called *minsit* and *maxsit*, respectively, previously defined by each airport in terms of its infrastructure and of the operating characteristics of each aircraft type;

3. The beginning of a duty period must occur at least 30 minutes before the scheduled departure time of its first flight (*brief time*);

4. The ending of a duty period must occur at least 30 minutes after the scheduled arrival time of its last flight (*debrief time*);

5. The elapsed time of a duty period, including brief and debrief times, must be at most of 11 hours (*maxelapse*);

6. The total number of flying time and landings in the same duty period are limited to 9 ½ hours and 5 landings, respectively. Airlines that make use of conventional aircrafts or turboprops can add 4 landings to the prescribed limit;

7. Between two duty periods an overnight rest period of 12 hours must be assigned;

8. Each crew member must return to its crew base in a maximum of 6 days;

9. Each crew member must receive at least 8 days off per month in its crew base, with 2 consecutive days off on a weekend (one on Saturday and the other on Sunday) and at least 1 day off on a week;

10. The total flying time of all duty periods assigned to each crew member must not exceed, in each month, quarter or year, respectively, for conventional aircraft, 100, 270 and 1,000 hours; for turboprop aircraft, 100, 255 and 935 hours; for jet aircraft, 85, 230 and 850 hours;

11. The total work time (total time of flying and ground service) of each crew member must not exceed 44 hours per week and 176 hours per month;

12. The crew members receive a fixed salary for 54 flying hours per month (minimum guarantee) and an additional remuneration for each exceeding flying hour;

13. As a quality criterion, the total flying time should be balanced among the crew members, aiming at the equalization of salaries.

2.1. Literature review

The CSP is usually divided in the literature into two subproblems, formulated and solved sequentially: Crew Pairing Problem (CPP) and Crew Rostering Problem (CRP). The CPP seeks to provide an optimal set of pairings that covers all the planned flights. Then, in the CRP, the best combination of rosters (composed by the pairings of CPP and other pre-defined activities) to crew members is determined, seeking the optimal coverage of planned flights and, eventually, the balancing of the total flying time among the crew members (Andersson et al., 1998; Barnhart et al., 2003; Kohl and Karisch, 2004; Gopalakrishnan and Johnson, 2005). Pairing (or crew rotation) is the work accomplished by crew member starting and ending at the same crew base, featuring a cycle. A pairing can be formed by one or more duty periods, which are series of sequential flights comprising a day's work for a crew member.

Approaches for the sequential solution of the CSP are based on the "generate and optimize principle". The subproblems CPP and CRP are usually modeled as a set partitioning (or covering) problem, and solved into two phases. First, all legal pairings (or legal rosters in the CRP) are generated and their costs computed, and then the best subset of these pairings is selected. The rules and regulations are applied only in the generation phase, which is implicit in the model (Barnhart et al., 2003; Kohl and Karisch, 2004; Gopalakrishnan and Johnson, 2005).

The explicit enumeration of all pairings (or rosters) can be difficult because of both the numerous work rules that must be checked to ensure legality and, more importantly, the huge number of potential pairings. In fact, for most real instances, explicit enumeration of pairings and rosters is not possible (Barnhart et al., 2003).

The depth-first search and the shortest path with resource constraints are usually applied in the generation phase of the CPP and the CRP. In the optimization phase, strategies based on heuristics or metaheuristics have been adopted, as presented by Cabral et al. (2000), Chang (2002), Lucic and Teodorovic (2007), Gomes (2009). The exact methods can be applied only to small problems (Gomes and Gualda, 2008).

The decomposition of the CSP into two subproblems (CPP and CRP) reduces its solution complexity. However, it does not lead to a real estimate of cost and influence the final solution quality. In CPP, the total cost of the selected pairings is minimized, but the real cost of the crew schedule can only be calculated after the assignment of the pairings to the different crew members is accomplished, i.e. after the CRP is solved, since the crew members may receive a fixed salary for a given amount of flight hours per month and an additional salary for each excess flight hour. Moreover, the availability of the different crew members is not considered in the CPP solution. So, conflicts may arise during the assignment of rosters to the crew members in the CRP, originating extra costs and loss of quality of the final solution.

To avoid these drawbacks, the state of the art of CSP involves an integrated solution, with both subproblems (CPP and CRP) solved simultaneously.

Approaches for integrated solution of the CSP are still at an early stage of development in the literature. Zeghal and Minoux (2006) proposed the integrated solution of the CSP with two distinct phases, in which the rosters of the crew members are formed from the grouping of the duty periods (instead of the pairings) with other activities, such as day offs, training periods, medical exams, meetings, and other, thereby skipping the intermediate phase to obtain a pairing solution. Thus, in the first phase, all legal duty periods are generated from the planned flights. Then, the duty periods previously generated in the first phase are assigned to the crew members, taking into account the coverage of all flights and the satisfaction of all restrictions. Even for the largest real problems, complete enumeration of all duty periods turns out to be possible because the number of duty periods is of the same order of magnitude as the number of planned flights.

Zeghal and Minoux (2006) formulated the CSP as a large scale integer linear program, which incorporates aspects of the current regulation, the collective agreements and the crew members' availabilities, replacing the set partitioning and the set covering models. The model was solved by CPLEX 6.0.2, taking into account 20 real test problems provided by the Tunisian airline company TunisAir. Since finding feasible integer solutions turned out to be difficult for some instances even after 8 hours of computation, the authors proposed a heuristic approach based on a rounding strategy embedded in a partial tree-search procedure, replacing CPLEX. The heuristic method achieved a better trade off between solution quality and computational effort.

Souai and Teghem (2009) adopted a similar approach as Zeghal and Minoux (2006). In the first phase, all legal duty periods are generated, but in the second phase the assignment of duty periods to the crew members is optimized through a hybrid genetic algorithm. In this hybrid genetic algorithm, the crossover and mutation operators are applied alternately (instead of sequentially). In addition, two local search heuristics are used to improve the solution.

3. METHODOLOGY

The methodology proposed for the integrated solution of the CSP is divided into two phases (generation and optimization), inspired on the approaches proposed by Zeghal and Minoux (2006) and by Souai and Teghem (2009). Initially, all legal duty periods are formed through a depth-first search procedure applied to a flight network (generation). Then, a Hybrid Genetic Algorithm (HGA) is used to determine the best combination of duty periods to the crew members from an initial solution obtained with the aid of an integer linear programming model, considering the coverage of all flights exactly once, the rules and regulations, the crew members' availabilities, the balancing of total flying time among the crew members, and the minimization of the total cost crew (optimization).

Regarding the research of Souai and Teghem (2009), the HGA proposed in this paper incorporates new mechanisms in the heuristic of the initial population generation, in the crossover operator, and in the local search heuristic. In addition, the crossover and mutation operators are applied sequentially, and not alternately.

3.1. Duty periods generation

The legal duty periods are generated through a depth-first search procedure applied to a flight network. In the flight network, $G = (N, A)$, the nodes $i \in N$ represent the flights as well as a source $s \in N$ and a sink $t \in N$. The arcs represent the legal connections between flights. The source node has an arc incident $(s, i) \in A$ on each node $i \in N$. The sink node receives one arc incident $(i, t) \in A$ from each node $i \in N$.

A pair of flights will have a connection arc between them if the arrival airport of the first flight is the same as the departure airport of the second one and the interval between the two flights allows a feasible connection (considering the prescribed interval at *minsit* and *maxsit*) within a duty period. The depth-first search procedure starts at the source node (root) and explore all legal connections $(i, j) \in A$. The legal paths *s* – *t* in the flight network represent the duty periods.

The cost of a duty period is computed through expression (1) and equals the idle time cost of the crew member plus the overnight rest period cost.

$$c_d = \alpha \times \left[elapse - \left(bt + ft_d + dt \right) \right] + oc_c \qquad (1)$$

where c_d is the cost of duty period *d;*

α is the work cost per minute of a crew member;

$elapse$ is the maximum elapsed time allowed for a duty period (in minutes);

bt is the brief time (in minutes);

ft_d is the total flying time of the duty period *d* (in minutes);

dt is the debrief time (in minutes);

oc_c is the overnight cost in city *c*.

3.2. The hybrid genetic algorithm (HGA)

Genetic Algorithms (GA) are heuristics search procedures that utilize the concept of biological structure to natural selection and survival of the fittest. The GA begins with a set of solutions (individuals) called population, where each individual is represented by its chromosome (sequence of genes). Solutions from one population are taken and used to form a new population. This is motivated by a hope that the new population will be better than the old one. Basically, the GA procedure includes selection, crossover and mutation. Solutions (parents) which are then selected to form new solutions (offspring) are selected according to their fitness. The crossover operator exchanges some genes in a solution (parent) by the corresponding genes of the other to generate a new solution (child). The mutation operator randomly selected a child and changes their genes according to the probability of mutation. The mentioned procedures are being executed until the number of generations gets to a predefined value (Reeves, 2003).

The traditional GA can be combined with other heuristics or metaheuristics, in order to mitigate their weaknesses, such as long processing times and premature convergence to a local optimum. This combination denotes a Hybrid Genetic Algorithm (HGA).

Figure 1 shows the HGA procedure adopted in this paper. The termination condition of HGA considers a maximum number of generations, given by *MaxGen*, and in each generation *N* new solutions (offspring) are produced, where *N* is the population size. The mutation is applied to one of the solutions generated at the crossover, with probability *Pm*, and an improvement procedure (local search) is applied to the best solution produced at each generation.

1. Build the initial population ($Gen = 0$);
2. While ($Gen < MaxGen$) do
3.   Repeat
4.   Select two parents for reproduction (roulette wheel method);
5.   Perform crossover;
6.   Perform mutation, with the probability Pm;
7.   Apply repair heuristic to illegal offspring;
8.   Until (N offspring are created);
9. Evaluate fitness of offspring;
10. Apply local search to the best offspring;
11. Select new population ($Gen = Gen + 1$);
12. End While;

Figure 1 – The hybrid algorithm genetic (HGA) procedure

3.2.1. Notations

The following notations will be considered throughout this paper:

D: set of all legal duty periods ($d \in D$), generated in depth-first search procedure;

J: set of days of the considered planning horizon ($j \in J$);

K: set of crew members ($k \in K$);

K_j: set of crew members available to work on day $j \in J$ ($K_j \subseteq K$);

D_j: set of all legal duty periods that start on day $j \in J$ ($D_j \subseteq D$);

D_j^k: set of all legal duty periods that can be assigned to the crew member *k* on day *j*, satisfying all rules and regulations;

F: set of flights to be covered in the considered planning horizon ($i \in F$);

F_j: set of all flights that start on day $j \in J$ ($F_j \subseteq F$);

F_{dj}: set of covered flights by duty period $d \in D_j$ on day $j \in J$;

Fnc_{nj}: set of non covered flights by solution *n* on day $j \in J$;

Fsc_{nj}: set of over-covered flights by solution *n* on day $j \in J$;

$Pena_{nj}$: penalty of the solution *n* related to non covered flights and over-covered flights on day $j \in J$, given by $Pena_{nj} = |Fnc_{nj}| + |Fsc_{nj}|$;

$Pena_n$: penalty of the solution *n* related to non covered flights and over-covered flights, given by $Pena_n = \sum_{j \in J} Pena_{nj}$.

3.2.2. Chromosome encoding

The chromosome (solution or individual) is represented by a matrix $X = (x_{kj})_{mxn}$, where $m = |K|$ and $n = |J|$. A gene x_{kj} takes the value 0 if the crew member *k* is not assigned to any duty period on day *j* (free day), the value -1 if the crew member *k* is unavailable to work on day *j*, i.e., if to the crew member *k* on day *j* was pre-assigned other activity, such as day off, training periods, medical exams and others, and a positive integer value *d* representing the code associated to the duty period $d \in D_j$ on day *j* assigned to the crew member *k*.

Figure 2 shows a chromosome with $|K| = 4$ and $|J| = 7$. In this example, to the crew member 1 were assigned the duty periods 19, 22, 40 and 47 on days 2, 3, 4 and 5, respectively. In addition, the crew member 1 is unavailable to work on days 1 and 7, and available to receive any duty period on day 6. In contrast, the crew member 4 was not used in this solution.

Days

Crew members	1	2	3	4	5	6	7
1	-1	19	22	40	47	0	-1
2	5	13	-1	33	0	-1	62
3	1	-1	21	0	-1	50	61
4	0	0	0	-1	-1	0	0

Figure 2 – Chromosome example

The cost of a chromosome is computed through expression (2).

$$C_n = \sum_{k \in K} c_k y_k \tag{2}$$

where C_n is the cost of individual *n*;

c_k is the cost of the duty periods assigned to the crew member *k* (rosters);

y_k is equal to 1 if the crew member *k* is used in the solution *n*, and zero otherwise.

The cost of the duty periods assigned to each crew member $k \in K$ is computed through expression (3).

$$c_k = \alpha_1 + \max\left\{0, \left(\sum_{d \in D_k} ft_d\right) - MG\right\} \times \alpha_2 + \sum_{d \in D_k} c_d \tag{3}$$

where α_1 is the fixed salary of a crew member;

D_k is the set of duty periods assigned to the crew member *k*;

ft_d is the total flying time of the duty period *d*;

MG is the total flying time associated to the fixed salary of a crew member (minimum guarantee);

α_2 is the additional remuneration for each exceeding flying hour;

c_d is the cost of duty period *d*.

3.2.3. Initial population

The generation of the initial population (of *N* individuals) is divided into two steps, given the complexity to checking all the rules and regulations.

The first step consists of determining a set of duty periods $\overline{D}_j \subseteq D_j$, for each day $j \in J$, to cover all flights $i \in F_j$ exactly once with minimal cost. For this purpose, an integer linear programming model (4), based on set partitioning problem, was considered:

$$\overline{D}_j = \left\{ \begin{array}{lll} Min & \sum_{d \in D_j} c_d y_d & \\ s.a. & & \\ \sum_{d \in D_j} a_{id} y_d = 1 & & \forall\, i \in F_j \\ y_d \in \{0,1\} & & \forall\, d \in D_j \end{array} \right\} \quad \forall\, j \in J \tag{4}$$

where a_{id} is equal to 1 if the flight *i* is covered by duty period *d*, and zero otherwise;

y_d is equal to 1 if the duty period *d* is included in the set $\overline{D}_j$, and zero otherwise.

The objective function seeks to minimize the number of selected duty periods, i.e. the number of crew members to work during day *j*.

The second step is addressed to generate a legal solution by assigning the duty periods of each set $\overline{D}_j$ to the available crew members ($k \in K_j$) on day *j*, day-by-day and pilot-by-pilot, ensuring the satisfaction of all rules and regulations. In this step, a constructive heuristic was used (see the pseudo-code in Figure 3).

In Figure 3, the selection order of the crew members (line 4) and duty periods (line 7), at each iteration, influences the balance of total flying times among the crew members, as in Cabral et al. (2000). Thus, three different strategies for selection of the crew members and duty periods were considered, to say, SCD-S1, SCD-S2 and SCD-S3.

The SCD-S1 strategy follows the approach proposed by Souai and Teghem (2009), where, at each iteration, a randomly selected crew member $k \in K_j$ receives the duty period $d \in D_j^k$ that covers the largest number of flights $i \in F_j$.

In the SCD-S2 strategy, the crew members $k \in K_j$ are initially sorted in an ascending order of priority assignment and total flying hours accumulated, and then sequentially selected for the assignment of randomly selected duty periods $d \in D_j^k$.

Therefore, in line 2 (Figure 3), the crew members $k \in K_j$ are sorted in an ascending order of priority assignment, considering two groups: first, the crew members who have already received some duty period in the solution, and second, the crew members not used in the solution. Then, the crew members of each group are reclassified in an ascending order of total flying hours accumulated. Thus, the SCD-S2 strategy aims to reduce both the unbalanced flying times among the crew members and the number of crew members used in the solution.

1. For each day $j \in J$ do
2.   Build the sets K_j and F_j;
3.   While ($K_j \neq \{\}$ and $F_j \neq \{\}$) do
4.   Select a crew member $k \in K_j$ (SCD-S1, SCD-S2 and SCD-S3 strategies);
5.   Build the set D_j^k;
6.   If ($D_j^k \neq \{\}$) then
7.   Select a duty period $d \in D_j^k$ (SCD-S1, SCD-S2 and SCD-S3 strategies);
8.   Assign the duty period *d* to the crew member *k*: $x_{kj} \leftarrow d$;
9.   Remove the covered flights by duty period *d* of the set $F_j : F_j \leftarrow F_j \setminus \{F_{dj}\}$;
10.   Remove the duty period *d* of the set $\bar{D}_j$: $\bar{D}_j \leftarrow \bar{D}_j \setminus \{d\}$;
11.   End If
12.   Remove the crew member *k* of the set K_j : $K_j \leftarrow K_j \setminus \{k\}$;
13.   End While
14.   Update the number of non covered flights and over-covered flights on day $j \in J$;
15. End For

Figure 3 – Second step of the generation of initial population (constructive heurist)

In the SCD-S3 strategy, the selection of crew members occurs as in the SCD-S2 strategy. However, the selection of duty periods follows a procedure based on the construction phase of GRASP metaheuristic, proposed by Feo and Resende (1995). In this case, at each iteration, it is determined the set D_j^k (line 5) to the selected crew member *k* (line 4), and then built up a restricted candidate list (RCL) with the *p* duty periods (set D_j^k) that cover the largest number of flights $i \in F_j$. Finally, a duty period $d \in RCL$ (randomly) to be assigned to the crew member *k*, where $p = \left\lceil \frac{|D_j^k|}{2} \right\rceil$, is selected.

The constructive heuristic does not guarantee the coverage of all planned flights. In some cases the crew members can fly as passengers in a duty period. This type of flight is called a deadhead. Deadheads are typically used to reposition a crew member to a city where he is needed to cover a flight, or to enable the crew member to return to his home base at the end a duty period. Thus, the fitness of an individual *n* with $Pena_n > 0$, i.e. with non covered flights or over-covered (deadhead) flights is penalized (as described in Section 3.2.4 below).

3.2.4. Fitness function

The fitness function measures the quality of each individual in the population. Individuals with higher fitness or quality are selected for crossover and survival. The fitness function of an individual is defined by the expression (5), introduced by Souai and Teghem (2009).

$$FF_n = \frac{CT_{\max} - CT_n}{CT_{\max}} \qquad (5)$$

where FF_n is the fitness function of the individual *n*, with $FF_n \in [0,1]$;
CT_n is the total cost of the individual *n*;
$CT_{\max}$ is the largest total cost in the current population.

The total cost of an individual *n* is associated with the penalty related to non covered flights and over-covered flights, the cost of the individual *n* (expression (2)) and the balancing of flying hours among the crew members.

The expression (6), adapted from Souai and Teghem (2009), is used to calculate the total cost of each individual of current population.

$$CT_n = \beta_1 \times Pena_n + \beta_2 \times C_n + \sigma_n \qquad (6)$$

where CT_n is the total cost of individual *n*;
$Pena_n$ is the penalty of the solution *n* related to non covered flights and over-covered flights;
C_n is the cost of individual *n*;
σ_n is the standard deviation function of flying hours assigned to the crew members in the individual *n*.

The parameters β_1 and β_2 must be defined adequately to minimize hierarchically the three terms of the expression (6), i.e. the penalty first, then the cost and then the standard deviation function.

The value of the parameter β_1 must ensure that $\beta_1 \times Pena_n >> C_n, \forall\ n$. Thus, β_1 is calculated as follows: first, the inactive duty period cost is determined, i.e. the cost of a duty period in which the crew member does not fly; then a illegal solution is generated, where the inactive duty period is assigned to the all crew members $k \in K$, in each day $j \in J$, assuming there are not free days; and finally, the value of β_1 is defined by:

$$\beta_1 = c_{\max} \times |K| \qquad (7)$$

where $c_{\max} = \alpha_1 + c_d \times |J|$ is the maximum cost of illegal schedule (upper bound) assigned to a crew member *k*.

The value of β_2 is determined through expression 8:

$$\beta_2 = \begin{cases} \dfrac{A+B}{2} & \text{if } A \neq 0, \\ B & \text{if } A = 0. \end{cases} \tag{8}$$

where $A = \min_{n=1,\ldots,N \ s.a. C_n \neq 0} \left\{ \dfrac{\beta_1 \times Pena_n}{C_n} \right\}$ and $B = \max_{n=1,\ldots,N \ s.a. C_n \neq 0} \left\{ \dfrac{\sigma_n}{C_n} \right\}$.

3.2.5. Selection, Crossover and Mutation

The selection of two parents *X* and *Y* for reproduction is performed through a roulette wheel method. In this case, each individual of the population is associated to a number of sectors of the roulette wheel, according to their fitness. Next, a random number between zero and the sum of the current population fitness is selected. Individuals associated to the selected sector in the roulette wheel are recombined.

The crossover operator recombines the genetic information (genes) of selected individuals *X* and *Y* (parents), in order to obtain two new individuals X' and Y' (offspring). At this point, four different crossover strategies were considered, named as SC-MP (Simplified Crossover in Multiple Points) and PC-MP (Probabilistic Crossover in Multiple Points), as introduced by Souai and Teghem (2009); RC-MP (Random Crossover in Multiple Points), adapted from Souai and Teghem (2009); and RC-SP (Random Crossover in Single Point), as introduced by Chang (2002).

In the SC-MP strategy, a number *n* is randomly determined, with $1 \leq n \leq \min\{|K|,|J|\}$. Next, *n* distinct genes are randomly selected, so that two genes are not selected in the same row $k \in K$ or same column $j \in J$. Finally, the selected genes are swapped between parents *X* and *Y*, generating the offspring X' and Y' (see Figure 4).

In the PC-MP strategy, the random selection of *n* distinct genes is performed as in the SC-MP strategy. Next, the selected genes whose content do not violate the legality of any solution X' and Y' are automatically swapped. For other selected genes, the exchange will depend on the degree of illegality of the solutions, measured by the penalty of day *j*. More precisely (for example, in solution X'), if $Pena_{X'j} \leq Pena_{Xj}$, then the exchange is accepted; otherwise, the exchange is accepted with a probability *P* defined by: $P = \dfrac{1}{Pena_{X'j} - Pena_{Xj} + 1}$ (see Figure 4).

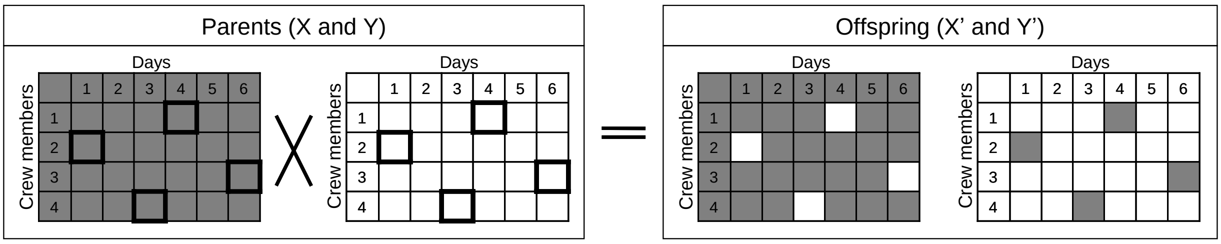

Figure 4 – SC-MP and PC-MP strategies

In the RC-MP strategy, a number *n* is determined at random, with $1 \le n \le \max\{|K|,|J|\}$. Next, *n* distinct genes are selected at random, with $x_{kj} \ne -1$ for any selected gene. Finally, only the selected genes are swapped between the parents *X* and *Y*. In this strategy, two genes can be selected in the same row $k \in K$ or same column $j \in J$ (see Figure 5).

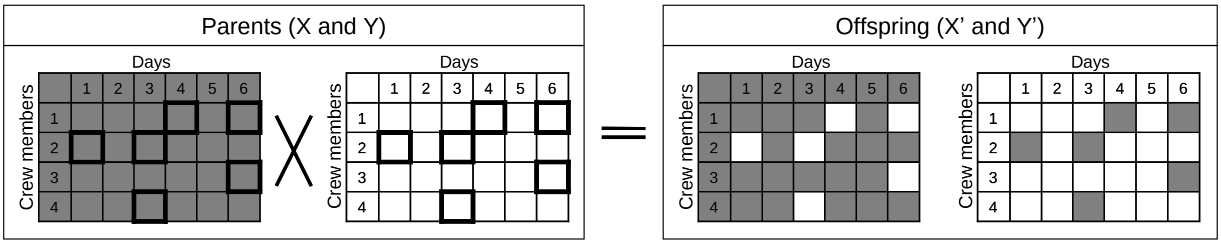

Figure 5 – RC-MP strategy

In the RC-SP strategy, a day $j \in J$ is selected at random. Then, the genes of the selected day are automatically swapped between the parents *X* and *Y* (see Figure 6).

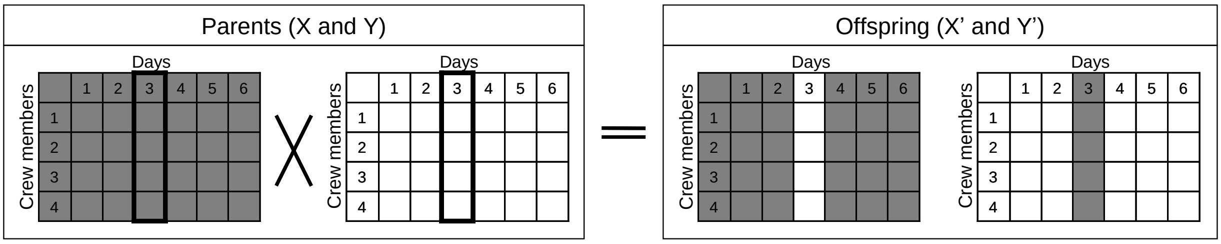

Figure 6 – RC-SP strategy

The mutation operator is applied to one of two created offspring at crossover, with probability *Pm*. First, it selects, randomly, one offspring (X' or Y'), a day $j \in J$ and two crew members k and k' from set *K*, such that $x_{kj} \ne -1$ and $x_{k'j} \ne -1$. Afterward, the genes x_{kj} and $x_{k'j}$ are swapped (see Figure 7).

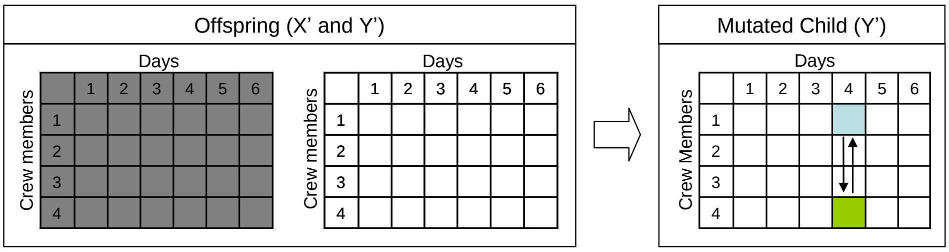

Figure 7 – Mutation operator

The legality of the solutions X' and Y' is not assured by crossover and mutation operators. For this reason, the repair heuristic will be applied afterward (see Section 3.2.6 below).

3.2.6. Repair heuristic

The repair heuristic follows immediately the application of the genetic operators (crossover and mutation) and is applied only to illegal offspring (X' and Y'), in order to correct the genes x'_{kj} with assignments that do not satisfy all the rules and regulations. Figure 8 shows the structure of the repair heuristic.

1. Let x'_{kj} be a illegal gene, where $x'_{kj} = d$ or $x'_{kj} = 0$;
2. Let F_{rh} be the set of flights to be covered in the repair heuristic;
3. For each illegal gene x'_{kj} do
4.   Build the set D_j^k;
5.   If ($D_j^k \neq \{\}$) then
6.   Build the set F_{rh}: $F_{rh} = Fnc_{nj} \cup F_{dj}$;
7.   If ($F_{rh} \neq \{\}$) then
8.   Assign the duty period $d \in D_j^k$ that covers the largest number of flights $i \in F_{rh}$ and the least number of flights $i \notin F_{rh}$ to the crew member *k*: $x'_{kj} \leftarrow d$;
9.   Else
10.   Assign the duty period $d \in D_j^k$ that covers the least number of flights $i \in F_j$ to the crew member *k*: $x'_{kj} \leftarrow d$;
11.   End If
12.   Else
13.   Restore the gene x_{kj} removed at crossover or mutation: $x'_{kj} \leftarrow x_{kj}$;
14.   End If
15.   Update the penalty of the individual X' ($Pena_{X'}$);
16. End For

Figure 8 – Repair heuristic

Note that, in line 8 (Figure 8), if $F_{rh} \neq \{\}$ then the duty period $d \in D_j^k$ that covers the largest number of flights $i \in F_{rh}$ and the least number of flights $i \notin F_{rh}$ is assigned to the crew member *k*, reducing the penalty of the solution *n* related to non covered flights and over-covered flights on day *j*. Otherwise ($F_{rh} = \{\}$) all flights $i \in F_j$ are covered and the gene x'_{kj} is equal to zero. Thus, the duty period $d \in D_j^k$ that covers the least number of flights $i \in F_j$ is assigned to the crew member *k* (line 10), reducing the penalty of the solution *n* related to over-covered flights on day *j*. When a legal duty period is not identified in the repair heuristic ($D_j^k = \{\}$), the gene x_{kj} removed during the crossover or mutation is restored (line 13). For example, $x'_{kj} \leftarrow x_{kj}$, i.e. the x'_{kj} is the illegal gene of offspring X' and x_{kj} is the legal gene of parent *X*. Thus, the legality of the solution at the end of the repair heuristic is ensured.

3.2.7. Local search

The local search, also known as neighbourhood search, is an improvement procedure, where the neighbourhood of the current solution is explored, at each iteration, in search of a better solution. In HGA, a local search is applied to the best offspring (solution *s*) produced at each generation. Thus, given a solution *s*, two neighbouring solutions s' are explored through two distinct movements: the reassignment movement of a duty period and the exchange movement of two duty periods. If one of the neighbouring solutions s' is better than the solution *s*, then it replaces *s* by s' ($s \leftarrow s'$). The illegal solutions s' are discarded.

The reassignment movement consists of removing a duty period assigned to a given crew member, and then of reassigning it to another crew member available on the same day. Figure 9 illustrates the reassignment movement, in which the duty period 21 is removed from the crew member 3 and then reassigned to the crew member 4 on day 3.

Solution *s*

Days (columns) / Crew members (rows)

	1	2	3	4	5	6	7
1	-1	19	22	40	47	0	-1
2	5	13	-1	33	0	-1	62
3	1	-1	21	0	-1	50	61
4	-1	0		31	44	51	-1

Solution *s'*

Days (columns) / Crew members (rows)

	1	2	3	4	5	6	7
1	-1	19	22	40	47	0	-1
2	5	13	-1	33	0	-1	62
3	1	-1	0	0	-1	50	61
4	-1	0	21	31	44	51	-1

Figure 9 – Reassignment movement

The exchange movement consists to swap the duty periods assigned to two crew members on the same day. Figure 10 illustrates the exchange movement, in which the duty periods 40 and 31 are swapped between the crew members 1 and 4 on day 4.

Solution *s*

Days (columns) / Crew members (rows)

	1	2	3	4	5	6	7
1	-1	19	22	40	47	0	-1
2	5	13	-1	33	0	-1	62
3	1	-1	21	0	-1	50	61
4	-1	0	0	31	44	51	-1

Solution *s'*

Days (columns) / Crew members (rows)

	1	2	3	4	5	6	7
1	-1	19	22	31	47	0	-1
2	5	13	-1	33	0	-1	62
3	1	-1	21	0	-1	50	61
4	-1	0	0	40	44	51	-1

Figure 10 – Exchange movement

In both movements, the selection of days, crew members and duty periods is done randomly.

3.2.8. New population

The update process of the population occurs at the end of each HGA generation, where the worst parents are replaced by better offspring. In this update process, it is important to avoid the occurrence of repeated individuals in the new population, in order to ensure diversity of the population and not premature convergence of the HGA to a local optimum.

4. TESTS AND PRACTICAL APPLICATIONS

The proposed methodology was tested to solve two instances of the CSP associated to the operation of a Brazilian airline, taking into account the schedule of the technical crew members (pilots) and the existence of only a crew base. Table I summarizes the test instances. Two coverage periods were considered: one of two weeks and one of four weeks, in order to better explore the performance of the proposed method.

Table I – Test instances

Instance Id	Aircraft	#Flights	#Crew Members (Pilots)	Planning Horizon (in days)	Coverage Period
MA1	Embraer 120 (Turboprop)	208	10	14	06/04/2008 to 19/04/2008
MA2	Embraer 120 (Turboprop)	416	10	28	01/06/2008 to 28/06/2008

The parameters used to calculate the cost of the duty periods (expression (1)) and the cost of the rosters (expression (3)) were: α = 2, $elapse$ = 660min, $bt = dt$ = 30min, oc_c = 200, α_1 = 2,000, *MG* = 54h and α_2 = 10.

The depth-first search procedure and the hybrid genetic algorithm (HGA) were implemented in C++, using the Microsoft Visual Studio C++ 6.0, and tested on a microcomputer PC Intel Core 2 Quad, 2.40 GHz, with 2GB of RAM under the Microsoft Windows XP (Professional) operating system. The mathematical model used in the first step of the generation of initial population (Section 3.2.3) was solved by linear programming package ILOG CPLEX 11.0.

Table II presents the results obtained in the phase of duty periods generation (Section 3.1), with CPU times of less than 1 second.

Table II – Results obtained in the phase of duty periods generation

Instance Id	#Flights	Flight Network		Duty Periods Generation	
		#Nodes	#Arcs	#Duty Periods	CPU Time (seconds)
MA1	208	210	620	868	<1
MA2	416	418	1,240	1,736	<1

Tables III and IV show the best HGA results (Section 3.2) to instances MA1 and MA2, respectively, after five independent runs, taking into account a maximum of 50,000 generations (*MaxGen*), a population of 200 individuals (after comparison with 100 and 300 individuals) and a probability of mutation *Pm* = 0.3% (after comparison with 0.1%, 0.5%, 0.7% and 1%).

In these tables, the first and second columns present the adopted strategy in the constructive heuristic for initial population generation (see Section 3.2.3) and at crossover (see Section 3.2.5), respectively. The following columns contain, respectively, the total solution cost (expression (6)), the penalty related to non covered flights and over-covered flights, the cost associated with the use of the crew members, the standard deviation of flying time assigned to the crew members, the number of crews members used in the solution, the number of the generation in which the solution was obtained, the CPU time (in seconds) and percentage deviation of the total cost in relation to the best solution. The values of the parameters β_1 and β_2 are shown at the bottom of the tables.

Table III – Results obtained at HGA for instance MA1

Initial Population	**Crossover Strategy**	**Total Cost** (CT_n)	**Penalty** ($Pena_n$)	**Cost** (C_n)	**Standard Deviation** (σ_n)	**#Crew Members**	**Generation**	**CPU Time (seconds)**	**CT (%)**
SCD-S1	SC-MP	224.93	0	98,520	33.41	10	50,000	211	53.36%
	PC-MP	208.00	0	96,250	20.88	10	50,000	178	41.82%
	RC-MP	203.31	0	97,020	14.69	10	10,000	56	38.62%
	RC-SP	207.96	0	97,830	17.77	10	30,000	94	41.79%
SCD-S2	SC-MP	188.84	0	69,950	52.85	8	50,000	246	28.75%
	PC-MP	192.37	0	85,712	25.74	9	30,000	104	31.16%
	RC-MP	155.69	0	67,035	25.37	8	10,000	57	6.15%
	RC-SP	174.56	0	83,325	12.57	9	40,000	122	19.02%
SCD-S3	SC-MP	193.61	0	70,520	56.51	8	50,000	230	32.00%
	PC-MP	168.62	0	70,520	31.52	8	50,000	192	14.97%
	RC-MP	**146.67**	**0**	**69,280**	**11.98**	**8**	**50,000**	**263**	**0.00%**
	RC-SP	148.82	0	68.120	16.39	8	20,000	68	1.47%
β_1 = 216,000 e β_2 = 0.001944									

Table IV – Results obtained at HGA for instance MA2

Initial Population	**Crossover Strategy**	**Total Cost** (CT_n)	**Penalty** ($Pena_n$)	**Cost** (C_n)	**Standard Deviation** (σ_n)	**#Crew Members**	**Generation**	**CPU Time (seconds)**	**CT (%)**
SCD-S1	SC-MP	294.19	0	177,040	10.20	10	50,000	305	31.91%
	PC-MP	307.36	0	177,610	22.45	10	40,000	200	37.81%
	RC-MP	295.11	0	177,610	10.20	10	10,000	115	32.32%
	RC-SP	292.34	0	174,760	12.00	10	30,000	141	31.08%
SCD-S2	SC-MP	266.45	0	145,405	33.20	9	50,000	281	19.47%
	PC-MP	266.77	0	153,362	20.76	9	30,000	148	19.61%
	RC-MP	252.37	0	150,470	11.00	9	30,000	322	13.16%
	RC-SP	253.24	0	149,330	13.70	9	30,000	129	13.55%
SCD-S3	SC-MP	243.93	0	127,320	39.69	8	20,000	160	9.37%
	PC-MP	242.39	0	127,362	38.08	8	30,000	144	8.68%
	RC-MP	**223.03**	**0**	**122,540**	**26.46**	**8**	**40,000**	**459**	**0.00%**
	RC-SP	235.77	0	125,610	34.28	8	20,000	86	5.71%
β_1 = 412,000 e β_2 = 0.001604									

Note that the SCD-S3 strategy (deterministic selection of crew members and random selection of duty periods on a restricted candidate list) combined with the random crossover in multiple points (RC-MP) produced the best final solution in both instances (MA1 and MA2). In contrast, the worst solutions were obtained with the SCD-S1 strategy (random selection of the crew members and deterministic selection of the duty periods), introduced by Souai and Teghem (2009).

5. CONCLUSIONS

This research treated the Crew Scheduling Problem (CSP), important part of the airlines operational planning. A methodology for the integrated modeling of the CSP is adopted, which eliminates the need to solve the Crew Pairing Problem (CPP). The rosters of the crew members are formed from the grouping of the duty periods (instead of the pairings) with other activities (such as day offs, training periods, medical exams, meetings, and others), leading to a final schedule with better quality.

For this purpose, a hybrid genetic algorithm (HGA) associated with a depth-first search was developed. The proposed methodology permitted to obtain feasible and efficient solutions for the considered instances, with reduced CPU times (order of 1 to 8 minutes). The results of tests and practical applications (see Tables III and IV) indicate that the SCD-S2 and SCD-S3 strategies, proposed in this paper for application of the constructive heuristic (see Section 3.2.3), were more effective than the SCD-S1 strategy, adopted by Souai and Teghem (2009). In addition, the RC-MP strategy also proposed in this paper was more effective than other crossover strategies (SC-MP, PC-MP and RC-SP). These results suggest that the innovations introduced in this research contribute to the state of the art of modeling the Crew Scheduling Problem (CSP).

Although tests have only considered a crew base, the proposed methodology can be used in case of multiple crew bases, becoming necessary to specify the crew members associated with each crew base.

It is important to note that the Brazilian air legislation considered in defining the restrictions of the CSP (see Section 2) is in line with the safety rules, labor regulations and collective agreements adopted internationally, which allows the proposed model to be adapted for the solution of CSP of other countries.

ACKNOWLEDGMENTS

The authors acknowledge the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and the Laboratório de Planejamento e Operação de Transportes (LPT / EPUSP) for supporting this research.

REFERENCES

ANAC (2009) Lei do Aeronauta (7.183/84). Available from <http://www.anac.gov.br/biblioteca/leis/lei7183%20.pdf>. Agência Nacional de Aviação Civil, Brasília, DF, Brazil.

Andersson, E., E. Houses, N. Kohl and D. Wedelin (1998) Crew pairing optimization. In: Operations Research in Airline Industry (G. Yu, ed.), Chapter 1, p. 1-31. Kluwer Academic Publishers, Boston.

Barnhart, C., A. Cohn, E. Johnson, D. Klabjan, G. Nemhauser and P. Vance (2003) Airline crew scheduling. In: Handbook of Transportation Science (R. W. Hall, ed.), p. 1-48. Kluwer Scientific Publishers, Boston.

Cabral, L. A. F., M. J. Freitas, N. Maculan and R. C. V. Pontes (2000) An heuristic approach for large scale crew scheduling problems at Rio-Sul Airlines. In 40th International Symposium of the AGIFORS, Istanbul, Turkey. Available from <http://www.iceb.ufop.br/decom/prof/marcone/Publicacoes/RioSul_agifors.pdf>.

Chang, S. C. (2002) A new aircrew scheduling model for short-haul routes. Journal of Air Transport Management , 8, p. 249–260.

Feo, T. A. and M. G. C. Resende (1995) Greedy randomized adaptive search procedures. Journal of Global Optimization, 6, p. 109-133.

Gomes, W. P. (2009) Programação de tripulantes de aeronaves no contexto brasileiro. Dissertação de Mestrado, Departamento de Engenharia de Transportes, Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brazil.

Gomes, W. P. and N. D. F. Gualda (2008). Otimização da formação de viagens no processo de alocação de tripulantes a aeronaves (crew pairing problem). Anais do XXII Congresso de Pesquisa e Ensino em Transportes, ANPET, Fortaleza, p. 1010-1020.

Gopalakrishnan, B. and E. Johnson (2005) Airline crew scheduling: State-of-the-art. Annals of Operations Research, 140, n. 2, p. 305–337.

Kohl, N. and S. E. Karisch (2004) Airline crew rostering: Problem types, modeling and optimization. Annals of Operations Research, 127, p. 223–257.

Lucic, P. and D. Teodorovic (2007) Metaheuristics approach to the aircrew rostering problem. Annals of Operations Research, 155, p. 311-338.

Reeves, C. (2003). Genetic algorithms. In: Handbook of Metaheuristics (F. Glover and G. A. Kochenberger, eds.), p. 55-82. Kluwer Academic Publishers, Boston.

SNA (2009) Convenção Coletiva de Trabalho – 2009/2011. Available from <http://www. aeronautas.org.br/index.php/documentos/70-convencao-coletiva-de-trabalho-aviacao-regular-20092011>. Sindicato Nacional dos Aeronautas, Rio de Janeiro, RJ, Brazil.

Souai, N. and J. Teghem (2009) Genetic algorithm based approach for the integrated airline crew-pairing and rostering Problem. European Journal of Operational Research, 199, p. 674–683.

Zeghal, F. M. and M. Minoux (2006) Modeling and solving a crew assignment problem in air transportation. European Journal of Operational Research, 175, p. 187-209.