International Journal of Industrial Engineering & Production Research

December 2013, Volume 24, Number 4
pp. 259-268

http://IJIEPR.iust.ac.ir/

ISSN: 2008-4889

Cockpit Crew Pairing Problem in Airline Scheduling: Shortest Path with Resources Constraints Approach

M. Reisi-Nafchi & Gh. Moslehi*

Mohammad Reisi-Nafchi, *Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran,*
Ghasem Moslehi *Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran*

KEYWORDS

Airline scheduling,
Crew pairing, set partitioning,
Column generation,
Shortest path with resource constraints

ABSTRACT

Increasing competition in the air transport market has intensified active airlines' efforts to keep their market share by attaching due importance to cost management aimed at reduced final prices. Crew costs are second only to fuel costs on the cost list of airline companies. So, this paper attempts to investigate the cockpit crew pairing problem. The set partitioning problem has been used for modelling the problem at hand and, because it is classified in large scale problems, the column generation approach has been used to solve LP relaxation of the set partitioning model. Our focus will be on solving the column generation sub-problem. For this purpose, two algorithms, named SPRCF and SPRCD, have been developed based on the shortest path with resource constraint algorithms. Their efficiency in solving some problem instances has been tested and the results have been compared with those of an algorithm for crew pairing problem reported in the literature. Results indicate the high efficiency of the proposed algorithms in solving problem instances with up to 632 flight legs in a reasonable time.

1. Introduction

The air transport industry has been the subject of extensive studies for many years because of the great number of complexities associated with it. Most problems in the field have been defined with the objective of increasing profitability and decreasing airline expenses. One of the most important of these problems is airline crew scheduling. This problem is important because crew costs rank second on the cost list of airlines after fuel costs [1]. In the literature of airline scheduling, the crew scheduling problem has been divided into crew pairing and crew rostering. This paper addresses the former and a more detailed treatment of the latter can be found in Belobaba et al. [2]. A distinction is commonly made in airline crew between cockpit crew and cabin crew. More importance is attached to the cockpit crew pairing problem due to such properties as their larger pays and the possibility of modelling them as one group on one day [3].

The following definitions are used in any description of the problem. In airline scheduling, a flight leg is a flight section from one airport to another with specified departure and arrival times. A duty period is a sequence of some flight legs in one day with small rest times in between and a pairing is a sequence of some duty periods with overnight rests in between. Each pairing starts and ends with the same crew and in the same crew base. So, the crew pairing problem is defined as finding pairings such that all flight legs are covered at minimum cost.

In the literature, the crew pairing problem is considered in terms of the schedule horizon time and the frequency of flight legs in one of three states: daily, weekly, and dated. In the daily problem, the assumption is that all flight legs are repeated each day of the week while, in the weekly problem, it is assumed that the flight schedule is repeated on a weekly basis.

* **Corresponding author: Ghasem Moslehi**
Email *moslehi@cc.iut.ac.ir*
Paper first received June. 06, 2012, and in accepted form March 03, 2013.

In the dated problem, usually the schedule horizon time is the month and flight frequency is irregular. More details can be found in Gopalakrishnan and Johnson [4].

Chu et al. [5] solved the daily crew pairing problem by considering a linear cost structure for each pairing. Their main objective was to propose a heuristic method for branching based on graph theory to solve large size problems. They applied this method to three problem sets of an approximate size of 1200 flight legs. Vance et al. [6] used column generation to solve the crew pairing problem in a branch and bound framework by employing a nonlinear cost structure to calculate each pairing cost. They used a multi-label shortest path algorithm to solve the sub-problem of column generation. Desaulniers et al. [7] formulated the crew pairing problem as an integer, nonlinear multi-commodity network flow problem with additional resource variables and then isolated the nonlinear aspects of the problem using Dantzig-Wolf decomposition. They finally solved the problem in the form of a set partitioning problem within a branch and bound framework.

Makri and Klabjan [8] developed a new pricing scheme for the column generation approach to the crew pairing problem. They proposed two exact and two approximate procedures to stop column generation in solving the airline crew pairing problem. Their approach which is based on a nonlinear cost structure for each pairing is applicable to the daily, weekly, and dated versions of the problem. Vance et al. [9] proposed a new formulation of the daily crew pairing problem using the nonlinear cost structure for each pairing. The main advantage of their formulation over the set partitioning model is its tighter linear lower bound as compared with the integral solution. However, solving the LP-relaxation of this formulation is more complex and time-consuming than that of the set partitioning model.

Yan and Tu [10] proposed a special network flow model for the cabin crew pairing problem and solved it by using the network simplex method. In fact, this model was developed for a Taiwanese airline that had simpler work rules for each pairing. Borndorfer et al. [11] used the set partitioning model with additional constraints to formulate the crew pairing problem.

Zeghal and Minoux [12] proposed an integer programming model for the crew pairing problem and used CPLEX to solve it. They did not consider the assumption of the cockpit crew being together in one day. AhmadBeygi et al. [13] developed an integer programming to generate pairings in the case of a nonlinear cost structure for pairings. Their objective was to produce a tool for testing new ideas in related problems. They claimed that their model could be solved by commercial softwares.

In this paper, the daily cockpit crew pairing problem is considered. To solve the problem, the set partitioning model is used and its LP-relaxation is solved using the column generation procedure. The main contribution of this paper is in developing a shortest path problem with resource constraints (SPPRC) for sub-problem of column generation and proposing two algorithms to solve it, when the cost structure of pairings is nonlinear. Also, comparing these two algorithms with a proposed one in the literature is another note in this paper.

The rest of the paper is organized as follows. In Section 2, definitions and assumptions of the problem are introduced. The solution approach to the problem is stated in Section 3 and two algorithms are developed in Section 4 for solving the sub-problem of column generation where also some special notes are made about these algorithms. The computational results of applying the proposed algorithms to some problem instances of two major Iranian airlines and their analysis are presented in Section 5. Finally, conclusions are presented in the last section.

2. Definitions and Assumptions

As mentioned in Section 1, this paper addresses the cockpit crew pairing problem on a daily basis assuming that no deadheading is allowed. There exist strong reasons to avoid deadheading in airlines because of its expenses and practical problems, especially in the daily problems. The concepts and assumptions used are as follows.

2-1. Imposed Rules

In the final solution of the crew pairing problem, all regional work rules and regulations, union requirements and other specific work rules of airlines must be duly observed. These rules are restrictive and help to decrease the size of the problem. However, they could equally make it more difficult to find an optimal solution. In this study, 12 common rules are considered as reported in Ho et al. [14]. These rules are shown in Table 1. Each duty period and pairing that considers the related work rules is referred as 'the legal duty period' and 'the legal pairing', respectively.

Tab. 1. Imposed rules in the problem

No.	Notation	Rule description
1	*MinSit*	Minimum sit time between flight legs
2	*MaxSit*	Maximum sit time between flight legs
3	*MinRest*	Minimum overnight rest time between duty periods
4	*MaxRest*	Maximum overnight rest time between duty periods
5	*MaxFly*	Maximum flying time per duty period
6	*MaxLeg*	Maximum flight leg number per duty period
7	*EarlyElapse*	Minimum elapsed time of a duty period
8	*MaxElapse*	Maximum elapsed time of a duty period
9	*EarlyDutyStart*	Earliest start time of a duty period
10	*Brief*	Brief time before each duty period
11	*Debrief*	Debrief time after each duty period
12	*MaxDuty*	Maximum duty period number per pairing

2-2. Cost Structure of Each Pairing

Based on a common method that has been used with slight differences in most recent studies in the literature ([4], [6], [8], [9], [13], [15-17]), the cost of each pairing, *p*, is equal to the maximum amount of three quantities calculated here from the following equation:

$$c_p = \max\left\{ f_p \times TAFB, \sum_{d \in p} c_d, mg_p \right\} \quad (1)$$

where, c_p is the cost of pairing *p*, c_d is the cost of duty period *d*, *TAFB* is the time away from the crew base, and f_p is a fraction between 0 and 1. Also, mg_p is the minimum guarantee payment for pairing *p* without any attention to its elapsed time. The cost of each duty period, *d*, is calculated using the following equation:

$$c_d = \max\left\{ f_d \times elapse, fly, mg_d \right\} \quad (2)$$

where, *elapse* designates the elapsed time of the duty period, *d*; f_d is a fraction between 0 and 1, *fly* designates the flying time of the duty period, *d*; and mg_d denotes the minimum guarantee payment for the duty period, *d*.

The values of f_p and f_d could be different for various airlines. It is notable that all of the above quantities are in time credit and the amount of the pairing cost could be easily converted to a given currency by multiplying it by a currency factor.

2-3. Revised Flight Network and Duty Period Network

To show feasible pairings in the crew pairing problem, two types of network, namely flight network and duty period network, have been introduced in the literature. Barnhart et al. [17] described these classic networks. In this paper, two new types of network are designed based on the classic types and designated as "revised flight network" and "revised duty period network". These networks are designed based on activity on node networks (AON), but the classic ones are based on activity on arrows (AOA).

The main advantage of these revised networks over their classic counterparts is the fewer number of nodes and arcs required. So, the proposed algorithm based on the networks designed in this work requires less memory and fewer calculations. The properties of these networks will be described below.

In the revised flight network, a flight node is considered for each flight leg and some arcs are drawn to show the legal connection between flight legs. Two virtual nodes also exist in the network as origin and sink nodes to which are connected to the start and end of pairings, respectively. Table 2 presents the data for four flight legs and Figure 1 illustrates the related revised flight network for these flight legs over two days.

Tab. 2. Data on four flight legs

Notation	Origin	Destination	Departure time	Arrival time
a	Teh	Isf	8:00	10:00
b	Teh	Shi	9:00	11:00
c	Isf	Shi	11:00	12:00
d	Shi	Teh	13:00	15:00

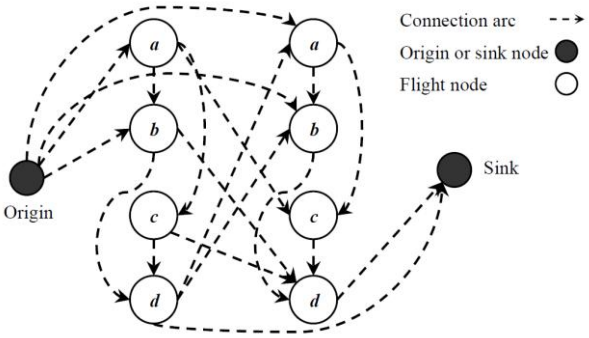

Fig. 1. Revised flight network

Similarly, in the revised duty period network, a duty node is considered for each duty period and some connection arcs are drawn in the network to illustrate the legal connections between duty periods. Also, two virtual nodes exist in the network as origin and sink which are connected to the start and end of pairings by two arcs.

3. Solution Approach

As mentioned in the introduction, the set partitioning model has been used in this paper to formulate the crew pairing problem. The mathematical programming model of the set partitioning problem is as follows:

$$\begin{aligned} \textit{Min} \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = 1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = 1 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = 1 \\ & x_j \in \{0,1\} \quad \textit{for} \quad j = 1,\ldots,n \end{aligned} \quad (3)$$

According to this model, each row or constraint represents a certain flight leg and each column or

variable represents a pairing. The value of each variable will be equal to 1 if the related pairing exists in the final solution; otherwise, it will be equal to 0. Variable j in the i^{th} row has a coefficient $a_{ij}=1$ if the flight leg i is covered by the pairing j; otherwise, $a_{ij}=0$.

Model (3) is a famous and powerful one in operations research. However, the main difficulty with the crew pairing problem is its large number of columns or legal pairings. So, obtaining all the variables in the above model is practically impossible. The column generation approach is used in this paper to solve the LP relaxation of the model (3).

In the column generation, the model is first limited to some initial variables, designated as the restricted master problem (RMP), and the problem is solved. All the variables are then implicitly considered and one or more new variables are added to the RPM in each iteration.

Finding new appropriate variables or columns for adding to the RPM is the duty of a sub-problem, which is called the pricing problem. If the sub-problem can not generate a new column, the solution process will stop.

The condition for selecting a new column to add to the RPM is that its reduced cost must be negative. In this paper, two algorithms based on the shortest path algorithms with resource constraints are proposed to solve the sub-problem in the column generation procedure. Below is a more detailed description of the algorithms.

4. Proposed Algorithms

In order to solve the sub-problem of column generation and to find pairings or paths with a negative, reduced cost, two shortest path algorithms with resource constraints are proposed in this section, which are based on the nonlinear cost structure mentioned in Section 2.2.

4-1. SPRCF Algorithm

Here, an algorithm named the shortest path with resource constraint in revised flight network (SPRCF for short) is proposed in the revised flight network to solve the problem. According to this algorithm, a label set with 9 labels similar to those in Table 3 is considered for each node in the revised flight network that is meant to account for the work rules in Section 2.1.

Using these labels, all the mentioned work rules can be modelled. However, SPRCF has the ability to take new rules into account by adding new labels to the label set if necessary. Each label set can be considered as a related vector of the form $(a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9)$.

Tab. 3. j^{th} label set of node i in the revised flight network

No.	Notation	Description
1	sig_dual_{ij}	Sum of dual values up to node *i*
2	$TAFB_{ij}$	Time away from crew base up to node *i*
3	DPC_{ij}	Cost of the current duty period which includes node *i*
4	$pred_{ij}$	Predecessor of node *i*
5	Fly_{ij}	Sum of flying time up to node *i* in the current duty period
6	$LegNo_{ij}$	The number of flight legs up to node *i* in the current duty period
7	$DutyNo_{ij}$	The number of duty periods from the origin node to node *i*
8	$Elapse_{ij}$	Elapsed time up to node *i* in the current duty period
9	$DutyStart_{ij}$	Start time of the current duty period

The following are the notations used in describing SPRCF:

n : sink node number in the network

$time(a,b)$: a function whose output is equal to the time length from time *a* to time *b*.

$size(k)$: the number of label sets of node *k*

day_k : a day number that includes node *k*

$fl(k)$: the flight leg related to node *k*

T_ori_i : departure time of flight leg *i*

T_dest_i : arrival time of flight leg *i*

$flying_i$: flying time of flight leg *i* calculated from the following equation:

$$flying_i = time(T\_dest_i, T\_ori_i) \quad (4)$$

$i_{du-start}$: the first flight leg of the duty period du

i_{du-end} : the last flight leg of the duty period du

$Brief$: brief time before each duty period

$Debrief$: debrief time after each duty period

$dutystart_{du}$: the start time of the duty period du which is equal to:

$$dutystart_{du} = T\_ori_{i_{du-start}} - Brief \quad (5)$$

el_duty_{du}: the elapsed time of the duty period du which is calculated from the following equation:

$$el\_duty_{du} = time(dutystart_{du}, T\_dest_{i_{du-end}}) + Debrief \quad (6)$$

ε : the error value accepted as being the reduced cost negative

In the SPRCF algorithm, the nodes are first numbered and an initial label set is assigned to each node. Then by considering the first node, the labels of the subsequent nodes that could be connected to this node are updated based on a certain pattern. Therefore, this node is removed from the list and the next node in the sequence is considered. This procedure is repeated for all nodes in the network so that finally the reduced

costs of the paths are calculated based on these updated labels. The SPRCF steps for solving the sub-problem of column generation are as follows.

Step 1: Numbering the node
Sort the flight legs of each day first by their departure time and then by arrival time increasingly. Assign a node to each flight leg in the network. Designate the origin node number as 0 and start numbering the other nodes from 1 and number each node based on the above sorted sequence.

Step 2: Producing the initial network
Step 2.1: Connect the origin node with a connection arc to each node whose related flight leg starts in the crew base.
Step 2.2: Check all flight legs of one day to establish *Minsit* and *Maxsit* rules and repeat it for each day. If these rules are considered for each couple of flight legs, connect them with a connection arc.
Step 2.3: Check each flight leg for connecting to flight legs of the other days according to the *MinRest* and *MaxRest* rules. If these rules are considered, connect them with a connection arc.
Step 2.4: Connect each node whose related flight leg ends in the crew base with a connection arc to the sink node.

Step 3: Initialization
Step 3.1: Consider the vector $(0,0,0,-1,0,0,0,0,0)$ as the first label set for the origin node but the vector $(-\infty,+\infty,+\infty,-1,0,0,0,0,0)$ for the other nodes.
Step 3.2: Designate the active list of pairings as ALP and let $ALP=\phi$, $j=0$, $k=j$ and $s=0$.

Step 4: Let $k=k+1$. If $k\geq n$, go to step 7; if nodes *j* and *k* are connected in the initial network, then go to step 5; otherwise, go on to step 7.

Step 5: Updating the label sets of nodes
Perform steps 5-1 to 5-10.
Step 5.1: Let $s=s+1$. If $s>size(j)$, then go to step 7.
Step 5.2: For s^{th} label set of node *j* and *k*, if $day_j=day_k$, then go to step 5.3; otherwise, go to step 5.4.
Step 5.3: If all the following conditions hold, then go to step 5.5; otherwise, go to step 5.1.

$$\begin{cases} LegNo_{js}+1\leq MaxLeg & (7)\\ time(DutyStart_{js},T\_dest_{fl(k)})+Debrief\leq MaxElapse & (8)\\ Fly_{js}+flying_{fl(k)}\leq MaxFlying & (9)\end{cases}$$

Step 5.4: If all the following conditions hold, then go to step 5.5; otherwise, go to step 5.1.

$$\begin{cases} DutyNo_{js}+(day_k-day_j)\leq MaxDuty & (10)\\ time(DutyStart_{js},T\_dest_{fl(j)})+Debrief\geq EarlyElapse & \\ or\quad day_j\neq day_{fl(pred_{js})} & (11)\\ time(DutyStart_{js},T\_dest_{fl(j)})+Debrief\leq MaxElapse & (12)\end{cases}$$

Step 5.5: For s^{th} label set of node *j*, if nodes *j* and *k* are connected, then let y_{jk}, c_{1jk} and c_{2jk} be increasing in sig_dual_{js}, $TAFB_{js}$ and DPC_{js}, respectively.
Step 5.6: Let $t=1$.
Step 5.7: (Dominance rule 1) If the following conditions hold, then go to step 5.1.

$$\begin{cases}(TAFB_{js}+c_{1jk})-(sig\_dual_{js}+y_{jk})\geq TAFB_{kt}-sig\_dual_{kt} & (13)\\ (DPC_{js}+c_{2jk})-(sig\_dual_{js}+y_{jk})\geq DPC_{kt}-sig\_dual_{kt} & (14)\end{cases}$$

Step 5.8: (Dominance rule 2) Consider the following conditions:

$$\begin{cases}(TAFB_{js}+c_{1jk})-(sig\_dual_{js}+y_{jk})\geq TAFB_{kt}-sig\_dual_{kt} & (15)\\ (DPC_{js}+c_{2jk})-(sig\_dual_{js}+y_{jk})\geq DPC_{kt}-sig\_dual_{kt} & (16)\end{cases}$$

If these conditions hold, erase t^{th} label set of node *k* and go to step 5.7.

Step 5.9: Let $t=t+1$.
Step 5.10: If $t\leq size(k)$, then go to step 5.7; otherwise, go to step 6.

Step 6: If $day_j=day_k$, then go to step 6.1; otherwise, go to step 6.2.
Step 6.1: Let $l_4=j$, $l_5=Fly_{js}+flying_{fl(k)}$, $l_6=LegNo_{js}+1$, $l_7=DutyNo_{js}$, $l_8=Elapse_{js}+time(DutyStart_{js},T\_dest_{fl(k)})$ and $l_9=DutyStart_{js}$.
Step 6.2: Let $l_4=j$, $l_5=flying_{fl(k)}$, $l_6=1$, $l_7=DutyNo_{js}+(day_k-day_j)$, $l_8=flying_{fl(k)}+Brief+Debrief$ and $l_9=T\_ori_{fl(k)}-Brief$.
Step 6.3: Add the following label set to node *k*.

$$((sig\_dual_{js}+y_{jk}),(TAFB_{js}+c_{1jk}),(DPC_{js}+c_{2jk}),l_4,l_5,l_6,l_7,l_8,l_9)$$

Step 7: Let $j=j+1$, $k=j$ and $s=0$. If $j<n-1$, go to step 4; otherwise, go to step 8.

Step 8: Finding pairings with negative reduced costs
Step 8.1: Let $h=1$.

Step 8.2: If the destination of the flight leg $fl(h)$ is not the crew base, then go to step 8.5; otherwise, let $g=1$ and go to step 8.3.

Step 8.3: If $g > size(h)$ or $\left(\max\left\{f_p \times TAFB_{hg}, DPC_{hg}, mg_p\right\} - sig\_dual_{hg}\right) \geq -\varepsilon$, then go to step 8.5. Otherwise, by using label *pred* of node h, find its predecessor node and continue to get to a node with the value -1 for *pred* (origin node). In this case, all the considered nodes from the origin to node h illustrate a path. If a flight leg does not repeat more than once in this path, put the related pairing of this path in the ALP.

Step 8.4: Let $g = g+1$ and go to step 8.3.

Step 8.5: Let $h = h+1$ and if $h < n$, then go to step 8.2; otherwise, go to step 9.

Step 9: Return ALP as output.

Step 10: Finish.

As mentioned in steps 5.7 and 5.8, two dominance rules 1 and 2 are developed in this algorithm for updating the label sets of nodes. These dominance rules have been selected and used in this algorithm based on several experiments and testing different ideas.

4-2. SPRCD Algorithm

As mentioned in Section 4.1, SPRCF is implemented in the revised flight network. It is possible to rewrite this algorithm for the revised duty period network. Therefore, in this Section, an algorithm named the 'shortest path with resource constraint in the revised duty period network' (SPRCD) is described. However, as SPRCF and SPRCD are similar in most respects, only the differences are mentioned here.

Since SPRCD is to be implemented in the revised duty period network, all duty periods that could be produced from flight legs based on work rules of duty periods are initially generated to produce the related network.

As in SPRCF, the nodes are first numbered and an initial label set is assigned to them. Then by considering the first node, the labels of subsequent nodes that could be connected to this node are updated based on a certain pattern.

This node is, therefore, removed from the list and the next node in the sequence is considered. This procedure is repeated for all nodes in the network to calculate the reduced costs of the paths according to these updated labels.

Because all the work rules of duty periods in SPRCD are observed while the initial network is produced, fewer labels are necessary in this algorithm to consider other rules. Therefore, for each node in the revised duty period network, it will suffice to consider one label set with only four labels. Table 4 shows these labels.

Tab. 4. j^{th} label set of node i in the revised duty period network

No.	Notation	Description
1	sig_dual_{ij}	Sum of dual values up to node *i*
2	$TAFB_{ij}$	Time away from crew base up to node *i*
3	DPC_{ij}	Cost of the current duty period in which node *i* is included
4	$pred_{ij}$	Predecessor of node *i*

By effecting the above changes, the SPRCD steps could be rewritten straightforwardly. It is necessary to mention that two dominance rules used in SPRCF are also applied here and in the same way.

4-3. Complementary Issues

In this sub-section, some notes and complimentary issues are presented about SPRCF and SPRCD.

The work rules considered in this paper are some of the most common airline rules. However, SPRCF and SPRCD algorithms are capable of considering any other rule that might be necessary so that if new rules cannot be taken into account by using the existing labels, one or more labels are simply added. The value error ε for being accepted as the negative reduced cost of one pairing, which is used in step 8.3 of SPRCF and applied in SPRCD for the same reason, is theoretically equal to 0. However, due to the cumulative error in calculations, it is reasonable to consider a small value for it in practice. This value is one of the setting parameters of the algorithm and the value considered for this parameter will be mentioned in the next section. The condition in step 8.3 of SPRCF and the equivalent condition in SPRCD, which states that no flight leg must repeat in one pairing, are observed in order to avoid deadheading and their elimination would yield a different solution. To better understand the nature of the problem at hand, consider the four flight leg data in Table 2. It is clear from Figure 1 that the pairing $b \to d \to a \to c \to d$, which includes the duty period $b \to d$ on day one and the duty period $a \to c \to d$ on day two, covers all the flight legs of the problem. But in this pairing, the flight leg *d* is repeated twice. One problem this might cause is that two groups of crew will be necessary for implementing this pairing and because crew 1 must fly the duty period $b \to d$ on day one and crew 2 must fly the duty period $a \to c \to d$ on the same day, one of these groups will have to fly as passengers in order to replace the other crew at the place considered for flying the next flight leg. This is a case of deadheading and, as mentioned before, not allowed to occur. As already stated in sub-section 2.2, the cost structure of each pairing in this paper is nonlinear and equal to the maximum of three quantities, one of which is equal to the maximum of the other two. However, all these quantities are linear. In the SPRCF and SPRCD algorithms, one label is saved to trace each quantity. The main reason for this is that it is not obvious from which quantity the cost of path is obtained before the end of one path is reached. If dominance rules 1 and 2

are not applied in SPRCF and SPRCD, then each time one node is considered, one new label set is added to the connected nodes according to the linear expression presented above; so when the algorithm steps are exhausted, all possible statuses have been considered and the best solution has been identified. This requires a lot of computer memory, even for small problems. So, it is very effective to use the dominance rules in order to make the calculations practical. However, application of these dominance rules could result in loss of the optimal solution [13]. In addition, when the cost structure of each pairing is linear, the problem is converted to one that aims at finding the simple shortest path in the network, which is a famous problem easily solved by the existing powerful algorithms reported in the literature.

5. Computational Results

To implement the algorithms proposed in Section 4, some problem instances were used selected from the flight schedules of two major Iranian airlines. Table 5 shows the specifications of these problem instances. Here, the problems are divided into two groups based on the number of duty periods that can be generated. The problems in set 1 have fewer duty periods than those in set 2. All of these problem instances have one crew base.

Tab. 5. Specifications of problem instances

Set	Problem name	Flight leg number
1	P1	26
	P2	34
	P3	42
	P4	32
	P5	110
2	P6	172
	P7	122
	P8	154
	P9	460
	P10	632

All the algorithms considered in this paper were coded in C++ using the BCP project of COIN-OR website (http://www.coin-or.org), in the Linux environment. The LP solver used to solve the LP models was CLP which is an open-source solver of COIN-OR. The algorithms were then run on a Laptop with 2.5 GHz CPU Core 2 Duo T9300 and with 2 GB RAM.

The values assumed for the pairing cost parameters were the same as those in Gopalakrishnan and Johnson [4] and in Vance et al. [9]. So, the values of f_p, f_d, mg_p and mg_d were set to $\frac{4}{7}$, $\frac{2}{7}$, 5 and 3, respectively and the obtained cost structure was used for solving the problem instances.

The value for ε must be selected with greater care as when a big value is selected for this parameter, then certain pairings with small negative reduced costs might not be generated, which in turn leads to the low quality of the final solution. In contrast, if this value is reduced to zero, the reduced cost of pairings with small positive reduced costs might become negative, in which case it will be useless to add these pairings to the RMP. Therefore, various values were tested and 0.001 was selected as the value for ε in both SPRCF and SPRCD algorithms. The values used for work rules in this paper were adapted from Ho et al. [14] and Ahmadbeygi et al. [13].

They are reported in Table 6. Three value sets are observed in this Table. The values in sets 2 and 3 are more rigorous than those in set 1. The result is a decreased number of legal pairings. These values were used in large size problems, especially in the methods implemented on the duty period network in an attempt to use less memory and to make the problem more tractable. However, it need be mentioned that making tighter and more rigorous values of the rules had an inverse relation with problem feasibility. This is witnessed by the fact that it would be possible to find a solution for a problem when the values in set 1 were used while those in sets 2 and 3 would render the problem infeasible.

Tab. 6. Values selected for work rules

No.	Rule notation	Set 1	Set 2	Set 3
1	*MinSit*	25 min	25 min	25 min
2	*MaxSit*	4 hours	4 hours	4 hours
3	*MinRest*	9 hours	12 hours	9 hours
4	*MaxRest*	48 hours	48 hours	48 hours
5	*MaxFly*	8 hours	8 hours	8 hours
6	*MaxLeg*	6	6	6
7	*EarlyElapse*	1 hours	12 hours and 45 min	13 hours
8	*MaxElapse*	13 hours	13 hours	13 hours
9	*EarlyDutyStart*	0:00 AM	0:00 AM	0:00 AM
10	*Brief*	45 min	45 min	45 min
11	*Debrief*	15 min	15 min	15 min
12	*MaxDuty*	4	3	4

For the purposes of this study, the column generation procedure was started by considering some artificial variables in RMP. In this case, one pairing was considered for each flight leg to cover all the flight legs. However, for the related columns of these pairings, a big value was considered as the coefficient

of the objective function in the set partitioning model. By doing this, the model attempted to remove these variables from the basis when a feasible solution was achieved. In other words, for each infeasible pairing that included one flight leg, a large amount of cost was considered since the model did not tend to keep it in the basis.

The performance of the proposed algorithms was compared with that in a previous study. More specifically, the SPRCF and SPRCD computational results were compared with those of the algorithm proposed by Makri and Klabjan [8], henceforth abbreviated to MKA.

This comparison was done because of the similarities of their study and this paper, especially in the cost structure of the pairings and the other assumptions for the problem. In addition, it is noteworthy to mention that comparison of the proposed methods has been considered less in the literature.

Table 7 illustrates the results of implementing SPRCF, SPRCD, and MKA on problem set 1 and by setting rule values to the values of set 1. The column designated by "Rule set" refers to the number of value sets of the rules from Table 6. Under the column "Column generation iterations" are included the number of iterations that the column generation procedure had passed before the pricing problem had not found any column. Also, the number of columns generated in all iterations of the column generation procedure is included under the column "Number of generated columns". If after stopping the column generation any artificial variables existed in the basis, their number was included under the column "Number of remaining artificial variables in the basis".

Tab. 7. Computational results of SPRCF, SPRCD and MKA for problem set 1

Algorithm	Problem name	Rule set	Flight leg number	Duty period number	Column generation iterations	Number of generated columns	Number of remaining artificial variables in the base	Objective value (hours)	Solution time (second)	Integrality of final LP solution
SPRCF	P1	1	26	-	5	103	0	52.6	0.98	✓
	P2	1	34	-	4	133	0	63.5	1.20	✓
	P3	1	42	-	4	205	0	73.86	1.03	✓
	P4	1	32	-	7	262	0	64.1	1.62	✓
SPRCD	P1	1	26	398	3	1492	0	50.6	0.81	-
	P2	1	34	326	3	1254	0	63.5	0.47	✓
	P3	1	42	476	4	2325	0	65.2	1.49	-
	P4	1	32	1232	5	4519	0	67.2	8.50	-
MKA	P1	1	26	398	6	100000	0	66.2	181.33	-
	P2	1	34	326	7	120000	0	66.6	193.64	✓
	P3	1	42	476	6	100000	0	88.6	251.00	-
	P4	1	32	1232	-	-	-	-	-	-

Tab. 8. Computational results of SPRCF, SPRCD and MKA for problem set 2

Algorithm	Problem name	Rule set	Flight leg number	Duty period number	Column generation iterations	Number of generated columns	Number of remaining artificial variable in the basis	Objective value (hours)	Solution time (second)	Integrality of final LP solution
SPRCF	P5	1	110	-	10	2870	0	184.0	25.06	-
	P6	1	172	-	13	5719	0	283.5	58.43	-
	P7	1	122	-	11	3014	0	189.2	26.55	-
	P8	1	154	-	11	4444	0	276.2	45.03	-
	P9	1	460	-	20	22198	0	846.5	354.08	-
	P10	1	632	-	36	41320	0	1119.8	1370.30	-
	P5	2	110	-	16	1925	0	206.6	13.01	-
	P6	2	172	-	20	2767	0	293.6	29.02	-
	P7	2	122	-	22	1336	0	235.2	16.47	-
	P8	2	154	-	21	2243	0	303.5	24.61	-
	P9	2	460	-	40	18574	0	893.2	493.87	-
	P10	2	632	-	42	37340	0	1155.3	1401.04	-
SPRCD	P5	2	110	2124	6	10700	0	204.7	48.85	✓
	P6	2	172	6542	7	20340	2	2000000314.3	626.74	-
	P7	2	122	7944	6	33574	0	261.7	662.28	-
	P8	2	154	11500	-	-	-	-	-	-
	P9	2	460	611427	-	-	-	-	-	-
	P10	2	632	-	-	-	-	-	-	-
	P5	3	110	721	6	2713	0	271.0	8.84	-
	P6	3	172	2789	8	10913	0	441.6	250.02	-
	P7	3	122	1928	6	2921	0	364.1	61.29	-
	P8	3	154	3052	7	8585	0	439.4	287.45	-
	P9	3	460	173362	-	-	-	-	-	-
	P10	3	632	-	-	-	-	-	-	-
MKA	P5	2	110	2124	-	-	-	-	-	-
	P6	2	172	6542	-	-	-	-	-	-
	P7	2	122	7944	-	-	-	-	-	-
	P8	2	154	11500	-	-	-	-	-	-
	P9	2	460	611427	-	-	-	-	-	-
	P10	2	632	-	-	-	-	-	-	-
	P5	3	110	721	4	66523	0	256.5	465.45	✓
	P6	3	172	2789	5	83201	0	438.7	2890.30	-
	P7	3	122	1928	4	65988	0	360.2	1567.43	✓
	P8	3	154	3052	-	-	-	-	-	-
	P9	3	460	173362	-	-	-	-	-	-
	P10	3	632	-	-	-	-	-	-	-

If the value of this column was nonzero, it meant that no feasible solution had been found. The column titled "Integrality of final LP solution" shows whether the final LP solution of the column generation procedure has been integral or not. The symbol "✓"in this column means that an integral solution was found for the LP relaxation problem in the root node and that it was not necessary to search for an integral solution in a tree search framework.

According to Table 7, in most cases, SPRCD has a lower objective value in the problem set 1 than in the other two algorithms. As shown, MKA not only has the worst objective value among the three algorithms, but it also failed to solve the problem P4 because of inadequate computer memory. Also, the solution time of MKA was far more than the two algorithms proposed in this paper.

In Table 8, the results of implementing the three algorithms SPRCF, SPRCD and MKA on problem set 2, with more difficult problems, is shown. According Table 8, SPRCF could solve all the problems in the rule sets 1 and 2. MKA, however, failed to solve any problem in rule set 2 which were easier than those in rule set 1, and could solve three problems of the six problems in set 2 with very tight values of rule set 3. SPRCD also solved two problems in rule set 2 and four problems in rule set 3. In addition, this algorithm was not able to remove all the initial artificial variables from the basis in problem P6 and did not find any feasible solution to it. The results of SPRCD and MKA in the rule set 1 are not shown in Table 8, since these algorithms exhibit very low efficiencies in solving problems of this kind.

Based on the objective value column in Table 8, MKA has a better objective value than SPRCD in the three problems it solved. However, one important issue here is that its solution time was very large even for the very tight values of rule set 3. The importance of this point can be appreciated when the above algorithms are applied to a tree search to find the integral solution of the problem, where too much time spent in each node of the tree could result in low efficiency of the algorithm or its inability to solve the problem at all.

6. Conclusions and Future Work

In this research, the daily cockpit crew pairing problem was considered based on the assumptions that important work rules are observed, a nonlinear cost structure exists for each pairing, and no deadheading is allowed.

This problem was formulated as a set partitioning problem and the column generation procedure was then used to solve its LP relaxation. To solve the sub-problem of column generation, two algorithms of the type shortest path algorithms with resource constraints, designated as SPRCF and SPRCD, were proposed. SPRCF and SPRCD were implemented on the revised flight network and the revised duty period network, respectively. These two types of network were presented as modified versions of the classic types to increase the efficiency of the proposed algorithms.

The two proposed algorithms and an existing one reported in the literature, abbreviated as MKA, were implemented on some problem instances. The results showed that the proposed algorithms were capable of solving problems of up to 632 flight legs as different values for work rules.

In addition, the algorithms exhibited a lower sensitivity for finding solutions to the values of work rules compared with MKA. Solution time for SPRCF and SPRCD was far less than that for the other algorithm. Since these algorithms are used to solve the sub-problem of column generation and also because the column generation procedure is called back several times in a branch and price tree, the long time in the column generation procedure imposed a great challenge that could even make it impossible to solve the problem.

For future work, one could extend the dominance rules in the proposed algorithms in order to improve the quality of the solutions obtained. In addition, some other methods like constraint programming could be employed to solve the sub-problem of column generation to compare the efficiency of these methods with those of the proposed algorithms.

References

[1] Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., "*Airline Crew Scheduling Under Uncertainty*", Transportation Science, Vol. 39, No. 3, 2005, pp. 340-348.

[2] Belobaba, C., Odoni, A., Barnhart, C., The global airline industry. John Wiley & Sons, Chichester, 2009.

[3] Barnhart, C., Cohn, A., "*Airline Schedule Planning: Accomplishments and Opportunities*", Manufacturing and service operation management, Vol. 6, No. 1, 2004, pp. 3-22.

[4] Gopalakrishnan, B., Johnson, E.L., "*Airline Crew Scheduling: State-of-the-Art*", Annals of Operations Research, Vol. 140, 2005, pp. 305-337.

[5] Chu, H.D., Gelman, E., Johnson, E.L., "*Solving Large Scale Crew Scheduling Problems*", European Journal of operational research, Vol. 97, 1997, pp. 260-268.

[6] Vance, H.P., Atamturk, A., Barnhart, C., Gelman, E., Johnson, E.L., Krishna, A., Mahidhara, D., Nemhauser, G.L., Rebello, R., "*A Heuristic Branch-and-Price Approach for the Airline Crew Pairing Problem*", Technical report, Aurban University, 1997.

[7] Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M.M., Soumis, F., "*Crew Pairing at Air France*", European Journal of Operational Research, Vol. 97, 1997, pp. 245-259.

[8] Makri, A., Klabjan, D., "*A New Pricing Scheme for*

Airline Crew Scheduling", INFORMS Journal on Computing, Vol. 16, No. 1, 2004, pp. 56-67.

[9] Vance, P.H., Barnhart, C., Johnson, E.L., Nemhauser, G.L., "*Airline Crew Scheduling: A New Formulation and Decomposition Algorithm*", Operations Research, Vol. 45, No. 2, 1997, pp. 188-200.

[10] Yan, S., Tu, Y.P., "*A Network Model for Airline Cabin Crew Scheduling*", European Journal of Operational Research, Vol. 140, 2002, pp. 531-540.

[11] Borndörfer, R., Schelten, U., Schlechte, T., Weider, S., "*A Column Generation Approach to Airline Crew Scheduling",* Technical report, ZIB, Berlin, 2005.

[12] Zeghal, F.M., Minoux, M., "*Modeling and Solving a Crew Assignment Problem in air Transportation*", European Journal of Operational Research, Vol. 175, 2006, pp. 187-209.

[13] AhmadBeygi, S., Cohn A., Weir, M., "*An Integer Programming Approach to Generating Airline Crew Pairings*", Computers and Operations Research, Vol. 36, 2009, pp. 1284-1298.

[14] Ho, R.L., E.L. Johnson, T.L. Shaw., "*Modeling Tools for Airline Crew Scheduling and Fleet Assignment Problems*", In: Ciriani,T., Gliozzi, S., Johnson, E., Tadei, R. (eds.) Operational Research in Industry, pp. 1-24, MacMillan Press Ltd, London, 1999.

[15] Klabjan, D., Johnson, E.L., Nemhauser, G.L., Gelman, E., Ramaswamy, S., "*Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching*", Computational Optimization and Application, Vol. 20, No. 1, 2001, pp. 73-91.

[16] Klabjan, D., Johnson, E.L., Nemhauser, G.L., Gelman, E., Ramaswamy, E., "*Airline Crew Scheduling with Regularity*", Transportation Science, Vol. 35, No. 4, 2001, pp. 359-374.

[17] Barnhart, C., Cohn, A., Johnson, E., Klabjan, D., Nemhauser, G., Vance, P., "*Airline Crew Scheduling*", In: Hall, R., (ed.) Handbook of Transportation Science, pp. 517-560, Kluwer, Boston, MA, 2003.