A Mathematical Programming Approach to Airline Crew Pairing Optimization

Diana C. Flórez, Jose L. Walteros, Miguel A. Vargas, Andrés L. Medaglia, Nubia Velasco
Centro para la Optimización y Probabilidad Aplicada (COPA)
Departamento de Ingeniería Industrial
Universidad de los Andes

Abstract. In the ever increasing competitive environment of the airline industry, efficient personnel's planning is among the most challenging tasks and solely responsible for the largest impact on an airline's cost structure. The problem is theoretically appealing because it is computationally difficult due to the huge number of possibilities, and the amount of rules that determine crew planning. This work proposes a methodology to determine the most efficient and least costly way to serve the flights in an airline schedule during a weekly planning horizon considering layovers and deadheads. The strategy is composed of three optimization models: a binary programming model, a model based on the set partitioning problem, and a third model based on the set covering problem, the two latter solved via column generation. Computational results for the three models are presented using test instances built from a mid-sized Colombian airline.

Keywords: Airline Crew Pairing, Column Generation, Set Covering Problem, Set Partitioning Problem

1. Introduction

Personnel planning and scheduling is one of the most challenging tasks faced by an airline. Its importance derives from the economical impact that an efficient planning has on business profitability. On the other hand, its difficulty arises from its combinatorial structure, closely tied to the number of flights in the schedule and the fleet size, among other factors.

The set of activities geared towards planning and scheduling staff is known in the literature as the *Airline Crew Scheduling Problem* (ACSP). ACSP is one of the practical combinatorial problems that is well known as a hard problem from a computational point of view (Guo et al., 2006). The objective is to define crew tasks such that all flights from a flight schedule are served in the most efficient way considering rules that determine their feasibility (e.g., maximum flying time) and crew preferences (e.g., maximum number of consecutive work days). In the last years, the problem has become even more complex due to the rapid airline industry growth, increase in passenger demand (especially business trips), and the ever increasing competition in the low-fare market. This has led researchers and software companies to invest a great amount of resources in developing highly technical decision support tools based on optimization (Guo et al., 2006).

Airline operation planning can be divided in four hierarchical phases, where the products of one phase are input for the next (Barnhart, 2003). These phases are:

> *Schedule Planning*: Based on the passenger demand forecasts, the output of this phase is what is known as the *flight schedule*. This flight offer is determined by the flight's time, day, and frequency.
>
> *Fleet Assignment*: This phase matches the flights in the schedule with a specific type of aircraft, considering plane capacity and flight characteristics (i.e., domestic, international, overseas).

Aircraft Routing: This phase deals with scheduling aircraft maintenance operations so that the fleet meets the best working conditions and strict regulations and is timely available to meet flight demand.

Crew Scheduling: The flight schedule and assignment are used in this phase to determine which tasks are allocated to each crew member such that all flights are served meeting labor, operational, and government regulations.

Within the crew scheduling phase two sequential and interconnected subproblems are the *Airline Crew Pairing Problem* (ACPP) and the *Airline Crew Rostering Problem* (ACRP). The ACPP determines the minimum number of (anonymous) crews required to serve all flights in a planning horizon, considering labor, operational, and government regulations. The results for the ACPP feed the ACRP, where specific staff members are assigned to those anonymous crews taking into account additional information such as seniority, vacation, staff abilities and preferences, among others. This work focuses on the ACPP.

After fuel, crews are the most significant cost for an airline (Crawford et al., 2006). While fuel is an exogenous factor and leaves little or no room for internal cost reductions, crews can be efficiently managed thus lowering costs (Pavlopoulou, 1996). Therefore, any crew savings will have a significant impact in the airline's finances. On the other hand, to compete in the low-cost airline market, airlines require low fares and a wide flight offer, posing difficulties to their own planning operations.

Most large-scale worldwide airlines use specialized software for crew planning such as *Sabre® AirCentre Crew* from Sabre Airline Solutions, Jeppesen's *Carmen Crew Management Suite*, Goal Systems with *GoalPlane®*, and Navitaire with their *Planning Optimization Suite* to efficiently manage their resources and reduce costs. However, some of these optimization-based tools are cost prohibitive and often out-of-reach for a number of small or even mid-sized airlines. Consequently, there are many opportunities for researchers to propose and develop in-house solutions to plan airline operations.

The development of mathematical models and optimization techniques in the airline industry began in the 1950s (Ahmed and Poojari, 2008; Yu and Yang, 1998) and has continuously evolved with changes in airlines' complexity and dynamics (Sylla, 2000; Klabjan, 2005). Nowadays, an airline can operate thousands of flights per day facing last minute changes due to factors as diverse as weather, security, and mechanical failures.

Most of the models that have been proposed to solve the ACPP present mathematical formulations based on the *Set Partitioning Problem* (SPP) or the *Set Covering Problem* (SCP). However, these formulations are complex and difficult to solve (Barnhart, 2003). The huge number of decision variables associated with the combinatorial structure of the problem, implies the use of large-scale optimization techniques such as column generation. Under column generation the original problem is decomposed into a *master problem* and an *auxiliary problem* (*subproblem*). The latter is commonly formulated as a *Restricted Shortest Path Problem* (RSPP) where feasible pairings are generated and later used to find a solution to the SPP or SCP (master problem). However, the RSPP is also computationally challenging, so it often requires optimization schemes such as *branch-and-bound* or *dynamic programming*. An important body of research has been dedicated to find computationally efficient solutions to the RSPP (Pavlopoulou, 1996).

The subproblem is often solved by integer programming (IP) using a network representation of the ACPP. A path in the network represents a feasible set of crew tasks and the objective is to find the path with the most negative reduced cost. The feasibility of a path is determined by a set of (regulation and labor) constraints that define the network structure (AhmadBeygi, 2008). Some researchers have used the network structure to derive constrained shortest path algorithms based on *dynamic programming* (DP), where side constraints are used to handle non-linear costs, path feasibility, and other constraints that depend on the specific problem (Anbil, 1998; Klabjan, 2005). Even though DP could be more efficient than IP, small variations to the original problem (e.g., new regulations) are handled easier in IP than in DP. Consequently, both IP and DP are frequently embedded into solution schemes for the ACPP.

Three of the most common methods to solve the ACPP are *off-line column generation* (CG), *dynamic column generation* (DCG) and *branch-and-price*. In off-line CG, either the complete set of feasible pairings or a subset is explicitly enumerated and then the SPP or SCP is solved (Arabeyre et al., 1969). However, because the solution space grows exponentially with problem size, it is not unusual to find billions of pairings in a 300-flight problem. Therefore, because explicit enumeration can become prohibitive, it is mostly used for small-size problems. Alternatively, in DCG, the *linear relaxation* (LR) of the integer master problem is solved generating one or more feasible columns in each call to the subproblem using a pricing criterion (e.g., reduced cost). This way, the complete set of pairings is partially described and only attractive columns are generated to solve the master problem. The drawback in this case is that integer solutions are not guaranteed at the end of the CG technique (Lavoie, 1988). To overcome this difficulty, a branch-and-price approach uses the same idea as DCG, but performs CG at every node of the branch-and-bound tree. Even though this approach can be computationally expensive due to the huge number of nodes in the search tree, integer solutions are guaranteed. To reduce the computational burden in large-scale problems (i.e., thousands of flights), it is possible to use a heuristic version of *branch-and-price* that sacrifices optimality in lieu of computationally efficiency. For example, Vance et al. (1997) present a detailed description of a branch-and-price heuristic approach, in which the ACPP is modeled using two possible graphical representations of the problem (i.e., a *leg network* and a *duty period network*). In each of these networks, a RSPP is solved to determine feasible pairings. Near optimal solutions are obtained for large-scale instances (e.g., nearly 2000 flights) using several stopping criteria for the CG procedure at each node of the branch-and-bound tree (e.g., maximum number of calls to the subproblem).

Besides column generation approaches for the ACPP, some researchers have proposed heuristics and metaheuristics. Kornilakis and Stamatopoulos (1981) propose a two-phase procedure to solve the ACPP using a combination of depth-first search and genetic algorithms. In the first phase, legal pairings are generated from feasible duties, built from the set of legs. In the second phase, the pairings are used in an optimization procedure taking into account fixed crew costs and deadheading costs. This methodology splits the problem into two smaller problems, thus reducing the number of feasible pairings obtained by solving the problem directly, but sacrificing optimality. The method is tested in a 2100-leg instance from a Greek airline.

Levine (1996) formulates a SPP for the ACPP and solves it with a genetic algorithm (GA) enhanced with a local search heuristic. In this hybrid approach the GA works directly on integer solutions, while the local search adds the hill climbing ability to explore large neighborhoods. This algorithm was tested on 40 real world problems from a known test set and optimal solutions were obtained for half of the instances and solutions within 5% of optimality for other nine.

Ching Chang (2006) formulates a dynamic model for the ACPP for cargo activities based on a SCP. The dynamic component arises from deadheading, because seat availability for pilots in one pairing determines the cost and seat availability of a connecting pairing. The solution methodology was evaluated using instances from a Taiwan international airline considering five objectives.

Hoffman and Padberg (1993) propose a *branch-and-cut* approach in which the ACPP is modeled as a SPP and solved to optimality. The key feature in this solver is that the SPP includes any number of base constraints. These base constraints allow modeling certain airline regulations such as not exceeding the number of crews in a given personnel base. The branch-and-cut optimizer was tested on 68 large-scale real-world crew scheduling problems. The optimizer solved a 145-leg problem in only 37 minutes.

Marsten and Shepardson (1981) present four successful applications in which scheduling problems in airlines and public transportation are formulated as a SPP. The authors introduce the concept of *resolved legs* which are a sequence of legs that must be served as a unit, that is, crews cannot change planes. They report an application in a North American airline in which partitioning constraints were relaxed using Lagrangian relaxation and solved by the subgradient method. They report savings of USD $300.000 in one type of aircraft.

For further information on the ACPP, SPP and SCP formulations, network representations, and solution techniques the reader is referred to Barnhart (2003), Gopalakrishnan and Johnson (2005), and Ernst et al. (2004).

This work proposes a solution methodology for the ACPP based on a set of interrelated optimization models. It focuses on the real-world scenario of a mid-sized Colombian airline facing a rapid expansion process. The decision support tools derived in this work are applicable and extensible to similar mid-sized airlines operating in the low-fare market.

The rest of the document is organized as follows. Section 2 defines basic ACPP terminology. Section 3 formally presents the ACPP, while Sections 4 thru 6 present the proposed models. Section 7 reports the computational experiments for the proposed models based on real data from a mid-sized airline. Finally, in Section 8, we conclude and outline possible research extensions.

2. Terminology

This section provides a brief definition of the most important terms in ACPP used throughout the document.

Figure 1 illustrates an example of a flight schedule for a three-day planning horizon made up of 18 flights. Each flight is represented by a gray rectangle with the letter F and an ID number (e.g., F1). A *leg* is a flight in the schedule that contains information on the origin and destination cities, departure and arrival times, and day of the week. A group of *legs* allocated to a crew in a given day is known as a *duty* and it is shown in Figure 1 with a blue dotted-line rectangle (e.g., F1 and F2). A feasible duty requires the destination city of one leg to be the origin city of the connecting leg; the duty connections should meet allowable time windows; and the duty should meet a set of labor, operational, and government constraints, such as not exceeding a maximum flying time. Similarly, a set of *duties* is known as a *pairing* and represents the tasks of an anonymous crew during the planning horizon. A pairing is feasible if its duties are connected, meaning that the destination city of one duty is the origin city of the connecting duty within an allowable time window. A feasible pairing must also satisfy a set of constraints such as not exceeding a maximum number of landings, among others.

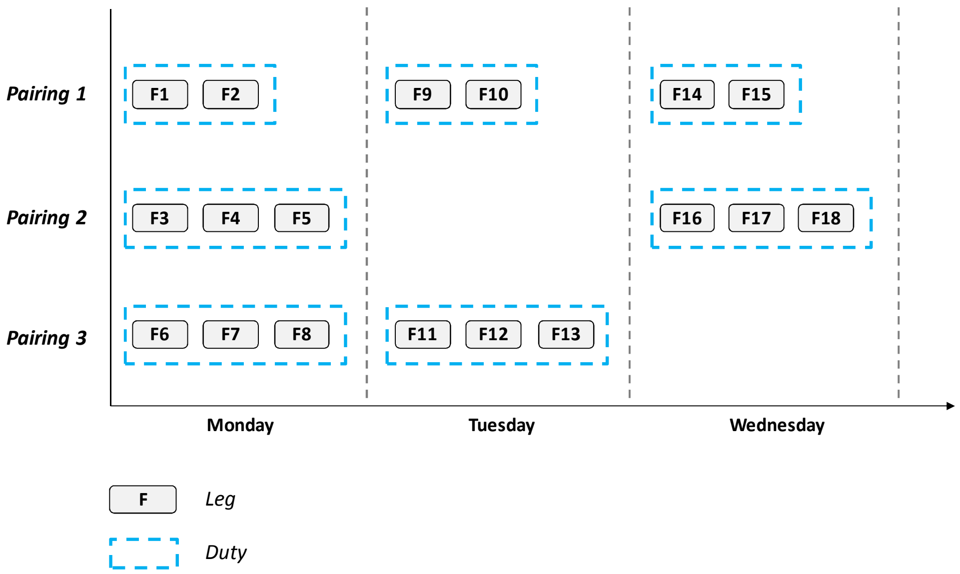

Figure 1. Example of a flight schedule

For a given pairing, its *flying time* is the total time a crew serves in the air its allocated flights. The *briefing time* is the time in advance a crew must show up at the airport before the first flight of a given duty, whereas the *debriefing time* is the time a crew stays in the airport after the last flight of its duty. Similarly, the *ground time* is the time a crew has to stay at the airport after any flight other than the last. *Service time* is the time a crew is doing work-related activities including briefing and debriefing. A *rest period* is the mandatory time that a crew must rest between two consecutive duties. Finally, *sit time* is the time that complements service time, that is, total service time minus flying, briefing, debriefing, ground, and rest times.

To illustrate these concepts, Figure 2 expands pairing 3 from Figure 1. Each flight contains origin and destination cities (e.g., C1 and C2 for leg F6), as well as departure and arrival times in minutes (e.g., 300 and 400 minutes for leg F6). The beginning and ending time for Monday operations is 0 and 1440 minutes, respectively. Thus, a departure time of 300 means that a flight departs at 05:00 a.m. on Monday, while a flight leaving at the same time on Tuesday would have a departure time of 1740 minutes. Based on the pairing in Figure 2, and assuming a 10-minute ground time at the destination city of each leg, debriefing times of 30 minutes, that all flights are domestic and require briefing times of just 60 minutes, and that rest time is at least 10 hours (i.e., 600 minutes); the flying, service, and sit times are 360, 1740, and 790, respectively.

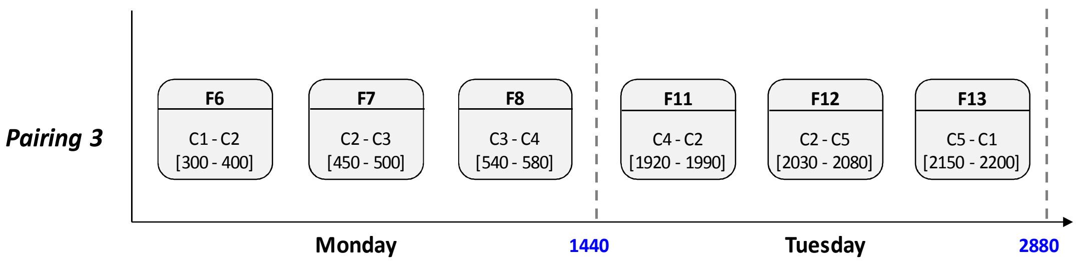

Figure 2. Example for pairing 3 in Figure 1

Pertaining crews, a *personnel base* is a city where the airline has a significant operation and where some crew members reside. Additionally, an *aggregated base* is used as reference to indicate the starting point for crew tasks. A *layover* is when, in the last flight of a duty, a crew lands in a city different from its initial base. In Figure 2, the crew has a layover on Monday at city C4 because that day it departed from the airport in city

C2. Finally, a *deadhead* is when a crew travels in a plane as passengers (not working) in order to serve in a connecting flight at the arrival city.

3. The Airline Crew Pairing Problem

The objective of the ACPP is to find the minimum number of crews required to cover all flights from a given flight schedule. To define pairings, we must consider a set of labor, operational, and government rules related to the company and the country where the airline operates. This work uses information from an airline that operates under the following conditions: 1) a pairing must begin and end its tasks at the same personnel base; 2) no pairing can exceed a maximum flying time, a maximum service time, a maximum number of landings, and a maximum number of duties; 3) a feasible pairing must satisfy several rules pertaining to a daily duty such as not exceeding a maximum flying time, a maximum service time, a maximum number of landings per day, and a minimum rest time (between duties).

Additionally, the way this particular airline operates makes it possible to separate and solve the problem by aircraft and crew type. Each crew member (pilot, copilot, and flight attendant) is trained exclusively to fly one type of plane. Therefore, the original problem can be solved for each of the three available types of aircrafts owned by the airline. On the other hand, even though a crew is comprised of pilots, copilots, and flight attendants, it can be separated in two groups: cockpit crew, which includes pilots and copilots, and cabin crew (flight attendants). This division is possible because a different set of rules applies to each crew type. Summarizing, the ACPP for the airline considered in this study can be solved separately by aircraft and crew type, thus allowing us to break the original problem into smaller parts and reducing its complexity.

The next sections present optimization models based on formulations for the ACPP: a Binary Programming model (ACPP/BP), a model based on the Set Partitioning Problem (ACPP/SP), and finally, a model based on the Set Covering Problem (ACPP/SC).

4. The Binary Programming Model (ACPP/BP)

The first model is based on a binary programming model formulation where the objective is to minimize the number of crews required to serve all legs from a flight schedule subject to a set of labor, operational, and government constraints. This model is based on the network representation of the ACPP shown in Figure 3.

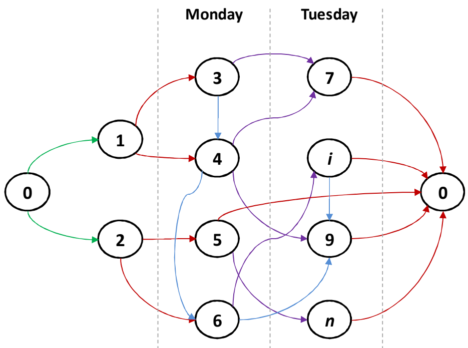

Figure 3. Example of the network representation for the ACPP

In this network, legs are represented by nodes and feasible flight connections are represented by arcs. Nodes 0, 1, and 2 are auxiliary nodes that represent the aggregated base where a pairing begins and ends, the beginning of a domestic flight, and the beginning of an international flight, correspondingly. Green arcs represent connections between the aggregated base and nodes 1 and 2. Red arcs connect nodes 1 and 2 with nodes whose origin cities are a personnel base; similarly, they connect nodes with the aggregated base if their destination city is a personnel base. Thus, we can identify the legs where pairings must begin and end. Blue arcs represent feasible daily connections between nodes, whereas purple arcs connect flights from different days, guaranteeing in both cases that the destination city matches the origin city and the departure time is at least the arrival time plus the ground time at the airport between connecting flights.

More formally, let G (N, A) be the directed acyclic graph composed of a set of nodes $N = \{0,\dots, i,\dots, n\}$ and the set of arcs A that represent feasible flight connections (i, j). The set $N_L=N \setminus \{0, 1, 2\}$ is comprised of nodes that represent legs from the flight schedule whereas the set N_L^c is its complement (base nodes). Let B be the set of cities that are personnel bases; S and I are the sets of domestic and international cities, respectively; D the set of days of the week; and C the set of crews. Let c_{max} be the maximum number of crews in the solution. For a pairing, t^{sp}_{max} is the maximum service time, t^{fp}_{max} the maximum flying time, and l^{p}_{max} the maximum number of landings. Similarly, for each duty, let d_{max} be the maximum number of duties in a pairing, t^{sd}_{max} the maximum service time, t^{fd}_{max} the maximum flying time, and l^{d}_{max} the maximum number of landings. Additionally, T is defined as the minimum rest time between consecutive duties. For each leg associated to node $i \in N_L$, let t_i, o_i, d_i, l_i, t^{d}_{i}, t^{a}_{i}, t^{f}_{i}, t^{g}_{i}, be the day of the week, origin city, destination city, number of landings, departure time, arrival time, flying time, and ground time associated with the airport of the destination city, respectively. It is assumed that $t^{d}_{i} = t^{a}_{i} = t^{f}_{i} = l_i = 0$, for all $i \in N_L^c$. Furthermore, let h_i be a binary parameter equal to 1 if the flight represented by node $i \in N_L$ begins in a national city, 0 otherwise; and let r_{iq} be a binary parameter equal to 1 if the flight represented by node $i \in N_L$ belongs to day $q \in D$, 0 otherwise. Finally, let t^{g}_{i} ($i \in N_L^c$) be the debriefing time ($i=0$) and briefing time at a national ($i=1$) and international city ($i=2$). Regarding the decision variables, let x_{ij}^k be a binary variable equal to 1 if the flight sequence (i, j) $\in A$ is served by pairing k, 0 otherwise. The mathematical formulation for the ACPP/BP follows:

$$\min \quad z = \sum_{k=1}^{c_{\max}} \sum_{\{ j\in N | (0,j)\in A \}} x_{0j}^k \tag{1}$$

subject to,

$$\sum_{k=1}^{c_{\max}} \sum_{\{ j\in N | (i,j)\in A \}} x_{ij}^k = 1 \qquad ; i \in N_L \tag{2}$$

$$\sum_{\{ j\in N | (i,j)\in A \}} x_{ij}^k - \sum_{\{ j\in N | (j,i)\in A \}} x_{ji}^k = 0 \qquad ; i \in N,\ k = 1,\dots,c_{\max} \tag{3}$$

$$\sum_{\{ j\in N | (0,j)\in A \}} x_{0j}^k \le 1 \qquad ; k = 1,\dots,c_{\max} \tag{4}$$

$$\sum_{\{ (i,j)\in A | o_j = b \}} x_{ij}^k - \sum_{\{ (j,0)\in A | d_j = b \}} x_{j0}^k = 0 \qquad ; i \in \{1,2\},\ k = 1,\dots,c_{\max}, b \in B \tag{5}$$

$$\sum_{\{ (i,j)\in A | i>0,\ j>2,\ t_i<t_j \}} x_{ij}^k \le d_{\max} \qquad ; k = 1,\dots,c_{\max} \tag{6}$$

$$\begin{aligned} & t_i^a + t_0^g - t_{\max}^{sd} - \left(\sum_{\{(m,n)\in A | t_m<t_i,\, t_n=t_i\}} \left(t_n^d - t_1^g \cdot h_n - t_2^g \cdot (1-h_n) \right) \cdot x_{mn}^k \right) + M \cdot x_{i0}^k \le M \qquad ; i \in N_L,\ k = 1,\dots,c_{\max} \\ & t_i^a + t_0^g - t_{\max}^{sd} - \left(\sum_{\{(m,n)\in A | t_m<t_i,\, t_n=t_i\}} \left(t_n^d - t_1^g \cdot h_n - t_2^g \cdot (1-h_n) \right) \cdot x_{mn}^k \right) + M \cdot x_{ij}^k \le M \qquad ; (i,j) \in A, i \in N_L, t_j > t_i, k = 1,\dots,c_{\max} \end{aligned} \tag{7}$$

$$\sum_{(i,\,j)\in A} t_i^f \cdot x_{ij}^k \le f_{\max}^{tp} \qquad ;k = 1,\ldots,c_{\max} \tag{8}$$

$$\sum_{(i,\,j)\in A} t_i^f \cdot r_{iq} \cdot x_{ij}^k \le f_{\max}^{td} \qquad ;q \in D, k = 1,\ldots,c_{\max} \tag{9}$$

$$\sum_{(i,\,j)\in A} l_i \cdot x_{ij}^k \le l_{\max}^{p} \qquad ;k = 1,\ldots,c_{\max} \tag{10}$$

$$\sum_{(i,\,j)\in A} l_i \cdot r_{iq} \cdot x_{ij}^k \le l_{\max}^{d} \qquad ;q \in D, k = 1,\ldots,c_{\max} \tag{11}$$

$$x_{ij}^k \in \{0, 1\} \qquad ;(i, j)\in A, k = 1,\ldots,c_{\max} \tag{12}$$

The objective function (1) minimizes the number of pairings required to cover all legs in a flight schedule. The set of constraints (2) guarantees that all legs are served exactly once. Constraints (3) guarantee crew flow through the set of legs. The constraints in set (4) indicate that each crew can leave the aggregated base node at most once to serve the first leg in the pairing. Constraint set (5) ensures that the city where a pairing starts and ends is not only the same but also a personnel base. The set of constraints (6) forces each pairing not to exceed the maximum number of duties allowed. Constraints (7) limit duty service time to its maximum value. Constraints sets (8) and (9) limit pairing and duty flying time, respectively. Likewise, constraints (10) and (11) limit the number of landings for pairings and duties. Finally, the binary nature of the decision variables is defined in (12). Henceforth the set of constraints (3)-(12) will be referred as the solution space Φ.

5. The Set Partitioning Model (ACPP/SP)

Let Ω be the set of all feasible pairings. Let a_i^k be a binary parameter equal to 1 if pairing p_k serves leg i, 0 otherwise. Let c^k be the cost of pairing $p_k \in \Omega$ and θ^k a binary decision variable equal to 1 if pairing p_k is part of the solution, 0 otherwise. The master problem for the ACPP/SP, denoted by MP(Ω), follows:

$$\text{MP}(\Omega): \qquad \min \quad z = \sum_{p_k \in \Omega} c^k \cdot \theta^k \tag{13}$$

subject to,

$$\sum_{p_k \in \Omega} a_i^k \cdot \theta^k = 1 \qquad ;i \in N_L \tag{14}$$

$$\theta^k \in \{0,1\} \qquad ;p_k \in \Omega \tag{15}$$

By fixing the cost $c^k = 1$ for each pairing, the objective minimizes the number of pairings (13) required to serve all flights exactly once. The constraint set (14) is known in the literature as the *partitioning constraints* because they guarantee that all legs are covered by exactly one pairing. Constraint set (15) defines the binary nature of the decision variables.

Note, however, that the mathematical formulation described by (13)-(15) assumes that all feasible pairings are known beforehand. Such an explicit enumeration of all feasible pairings is a dreadful process and can be computationally prohibitive even for moderate-size instances on the hundreds of legs.

To solve the problem, we propose a column generation (CG) approach. Instead of directly solving the MP(Ω), we solve a restricted master problem MP (Ω_1), where Ω_1 is a partial set of feasible pairings such that $\Omega_1 \subseteq \Omega$. In the auxiliary problem (subproblem), pairings that belong to Ω are generated using the reduced cost of the column associated with the pairing as the criterion for becoming part of MP (Ω_1). The feasibility of the generated pairing is guaranteed in the subproblem by enforcing the constraints (3)-(12) defining the solution space Φ. Finally, to solve the ACPP/SP via CG, the binary nature of the decision variables (15) is relaxed, thus obtaining the following linear programming formulation for the restricted master problem MP(Ω_1):

$$MP(\Omega_1): \quad \min \quad z = \sum_{p_k \in \Omega_1} \theta^k \tag{16}$$

subject to,

$$\sum_{p_k \in \Omega_1} a_i^k \cdot \theta^k = 1 \qquad ; i \in N_L \tag{17}$$

$$\theta^k \geq 0 \qquad ; p_k \in \Omega_1 \tag{18}$$

The dual problem of MP(Ω_1) is defined as D(Ω_1), where λ_i is the (unrestricted) dual variable associated with the *i-th* constraint from constraint set (17). The dual problem D(Ω_1) follows:

$$D(\Omega_1): \quad \max \quad z = \sum_{i \in N_L} \lambda_i \tag{19}$$

subject to,

$$\sum_{i \in N_L} a_i^k \cdot \lambda_i \leq 1 \qquad ; p_k \in \Omega_1 \tag{20}$$

$$\lambda_i \quad \text{unrestricted} \qquad ; i \in N_L \tag{21}$$

Note that a feasible pairing p_k that violates constraint (20) does not belong to Ω_1, but to $\Omega \backslash \Omega_1$; for this reason, p_k is a good candidate to belong to set Ω_1. Hence, the following subproblem tries to find a pairing that violates the set of constraints (20), that is, a column associated with a pairing with negative reduced cost.

$$\min \quad z = 1 - \sum_{\{(i,j) \in A \mid i \in N_L\}} \lambda_i \cdot x_{ij} \tag{22}$$

subject to,

$$\mathbf{x} \in \Phi \tag{23}$$

Note that in the model defined by (22)-(23), the parameter a_i^k from MP(Ω_1) is replaced by x_{ij}, a binary variable equal to 1 if arc (i, j) is used in the solution, 0 otherwise. These variables belong to the space Φ, that is, it considers all ACPP/BP constraints that define a feasible pairing except the one that guarantees that all legs are served exactly once.

The interaction between MP(Ω_1) and the subproblem works as follows. First, MP(Ω_1) passes the values of the dual variables λ_i to the subproblem, where they are used as parameters in the objective function (22). Once the subproblem is solved and it is possible to identify a feasible pairing with negative reduced cost, the set of legs that are served (built from the x_{ij}'s) are sent back to MP(Ω_1) as input for parameter a_i^k (one column). In the more general case that the pairing cost (c^k) depends on the structure of the generated pairing, this information would also be sent by the subproblem to MP(Ω_1). This communication process between the two problems is done iteratively until no further feasible pairings with negative reduced costs are found, that is, when there is no such pairing in $\Omega \backslash \Omega_1$.

The solution for MP(Ω_1) indicates which of the generated pairings in the subproblem will serve all legs in the least costly way. However, because the linear relaxation of MP(Ω) is being solved, an integer solution is not guaranteed. If the solution is not integer at the end of the CG, we run a *branch-and-bound* procedure with the set of generated columns (Ω_1) looking after an integer solution for the original problem. This branch-and-bound (rounding) process often finds a solution with a value greater than or equal to that of the linear relaxation; but in the case no integer solution is found, the linear relaxation found with CG is used as a lower bound for the original problem.

6. The Set Covering Model (ACPP/SC)

The third proposed model is based on the set covering formulation in which the number of pairings is minimized subject to serving all flights *at least* once. The ACPP/SC formulation follows:

$$\min \quad z = \sum_{p_k \in \Omega} c^k \cdot \theta^k \tag{24}$$

subject to,

$$\sum_{p_k \in \Omega} a_i^k \cdot \theta^k \geq 1 \qquad ; i \in N_L \tag{25}$$

$$\theta^k \in \{0,1\} \qquad ; p_k \in \Omega \tag{26}$$

The objective function (24) minimizes the cost of the required crews. The set of constraints (25) indicates that a leg can be served more than once, meaning that *deadheading* is allowed. Related to their planning operations, airlines may fix a cost to allow or penalize deadheads, such as the opportunity cost of not selling those plane tickets versus the cost of transporting the crew by other means. Including deadheading costs c^{dh}, the mathematical formulation for the ACPP/SC is modified as follows:

$$\min \quad z = \sum_{p_k \in \Omega_1} c^k \cdot \theta^k + c^{dh} \cdot \left(\sum_{p_k \in \Omega_1} \sum_{i \in N_L} a_i^k \cdot \theta^k - |N_L| \right) \tag{27}$$

subject to,

$$\sum_{p_k \in \Omega_1} a_i^k \cdot \theta^k \geq 1 \qquad ; i \in N_L \tag{28}$$

$$\theta^k \geq 0 \qquad ; p_k \in \Omega_1 \tag{29}$$

The objective function defined by equation (27) includes two types of costs: a fixed cost related to a crew's base salary (selecting a pairing in the solution) and a deadheading cost. The deadheading cost (second term in the objective function) is defined as the number of times a leg is served multiplied by the cost penalizing it. Finally, to use CG, the integer nature of the decision variables in (26) is relaxed (29).

Similarly to ACPP/SP, a new optimality condition in the ACPP/SC is derived and embedded in the following subproblem:

$$\min \quad z = \left(\left(c + c^{dh} \cdot \sum_{(i,j) \in A} x_{ij} \right) - \sum_{(i,j) \in A} \lambda_i \cdot x_{ij} \right) \tag{30}$$

subject to,

$$\mathbf{x} \in \Phi \tag{31}$$

Because the pairing cost is the same for all pairings, in equation (30) we define $c^k \equiv c$. The definition of this new cost is important because now deadheading costs are included, and so, it is not possible to simply fix $c^k = 1$ as in the ACPP/SP. This formulation requires a tuning of the magnitude of c^{dh} compared to c to allow or forbid deadheads. Finally, the space Φ satisfies all ACPP/BP constraints that define a feasible pairing except the one that guarantees that all legs are served exactly once.

Another common situation in airline planning operations is to allow crew *layovers*. To incorporate layovers in the ACPP/SC formulation, y_i is defined as a binary variable equal to 1 if there is a crew layover at the destination city of node i given that this node belongs to the generated pairing, 0 otherwise. Additionally, let c^p_i be the cost of a layover at the destination city of node i.

To connect decision variables x_{ij} and y_i we define a set of auxiliary variables. Let $\hat{e}$ be the city where a pairing begins and e_i the destination city associated with node i. If these two variables have the same value, there is no layover in the destination city associated with node i. Additionally, let α_i be a binary variable equal to 1 if there is a layover at the destination city of node i. The difference between α_i and y_i is that, in the first one, there is no guarantee that node i, where there could be a layover, is actually served by the generated pairing. The expressions to define $\hat{e}$ and e_i follow:

$$\hat{e} = \sum_{j \in N} \sum_{i \in N_L} o_j \cdot x_{ij} \qquad (32)$$

$$e_i = \sum_{j \in N} d_i \cdot x_{ij} \qquad ; i \in N_L, t_i < t_j \qquad (33)$$

Furthermore, the expressions that relate $\hat{e}$ and e_i with α_i are:

$$\begin{aligned} & e_i - \hat{e} + (1-\alpha_i) \cdot M \leq M \\ & \hat{e} - e_i + (1-\alpha_i) \cdot M \leq M \qquad ; i \in N_L \end{aligned} \qquad (34)$$

where $M >> 0$.

Therefore, the mathematical formulation, including deadheads and crew layovers, is:

$$\min \quad z = \left(\left(c + c^{dh} \cdot \sum_{(i,j) \in A} x_{ij} + \sum_{i \in N_L} c_i^p \cdot y_i \right) - \sum_{(i,j) \in A} \lambda_i \cdot x_{ij} \right) \qquad (35)$$

subject to,

(32),(33),(34)

$$\begin{aligned} & y_i \leq \alpha_i && ; i \in N_L \\ & y_i \leq \sum_{j \in N} x_{ij} && ; i \in N_L, t_i < t_j \\ & y_i \geq \sum_{j \in N} x_{ij} + \alpha_i - 1 && ; i \in N_L, t_i < t_j \end{aligned} \qquad (36)$$

$$\mathbf{x} \in \Phi \qquad (37)$$

$$y_i \in \{0,1\} \qquad ; i \in N_L \qquad (38)$$

$$\alpha_i \in \{0,1\} \qquad ;i \in N_L \qquad (39)$$
$$e_i \geq 0 \qquad ;i \in N_L \qquad (40)$$
$$\hat{e} \geq 0 \qquad (41)$$

7. Computational experiments

The proposed models were tested on flight schedules provided by a mid-sized Colombian airline that handles an average of 152 flights per day, both domestic and international. The airline covers the fast growing low-fare market segment, thus facing particular challenges. For instance, the airline recently launched 144 new routes to its flight offer, complicating operation planning even more. The company operates a fleet of 14 aircrafts of three types and a staff of 245 employees, between pilots, copilots, flight attendants, and support personnel. Currently, crew planning is manually done, that is, the pairing definition is hand-built. For a daily planning horizon, without considering deadheads or layovers, it can take nearly three hours to find a reasonable solution. Consequently, the proposed models ACPP/BP, ACPP/SP, and ACPP/SC were developed as an alternative to make pairing planning faster and more efficient, thus providing the airline with a tool to make it more competitive in the fierce low-cost airline market.

The models were evaluated using instances built from the current airline's flight schedule. The instances were named using the following format F_DD, where F represents the aircraft type and DD the collection of days of the week that define the planning horizon. For example, instance D202_MT is used for planning crew operations Monday through Tuesday for a Dash-8 202. The three models were solved using the commercial optimizer Xpress-MP® Version 18.10.00 on a Dell Optiplex 755 with an Intel Core(TM) 2 Duo processor running at 3.00GHz and 4GB of RAM memory.

Table 1 shows the results for the ACPP/BP model. The first column contains the name of the instance; the second column defines the instance size in terms of the number of legs; the third and fourth columns present the number of crews obtained with the linear relaxation (LR) of the original model and the integer model, respectively; and finally, the fifth column presents the computational time (in seconds).

Table 1: Results for the ACPP/BP

Instance	Legs	Required crews (LR)	Required crews (IP)	Integer sol?	CPU time (s)
D202_M	24	2	4	yes	2.41
D202_MT	48	4	4	yes	124.56
D202_MTW	72	4	4	yes	3557.02
D202_MTWT	96	4	4	yes	31427.63
D202_MTWTF	120	4	-	no	195.56
D202_MTWTFS	138	4	-	no	488.06
D202_MTWTFSS	152	4	-	no	554.78
D201_M	99	16	16	yes	89.36
D201_MT	198	12	-	no	3065.03
D201_MTW	297	-	-	-	-
D300_M	45	5	8	yes	10.53
D300_MT	89	8	-	no	8.57
D300_MTW	134	8	-	no	194.43
D300_MTWT	178	8	-	no	1039.74

Table 1 shows that optimal solutions were found for 6 out of 14 instances with a relative variability in computational times. For one-day instances (D202_M, D201_M, and D300_M) the solution matches the hand-built solution in terms of the number of crews needed. Due to the nature of the flight schedule, in which flights are the same from Monday through Friday for one of the aircrafts (D202), it was expected that solutions for instances associated with this type of plane would be the same. For other instances, only the linear relaxation was reported (e.g., D300_MT), after aborting a long execution of the branch and bound procedure without even finding the first integer solution. The remaining instances were not even solved because computational time significantly increased with problem size.

Table 2 presents the results for the second model, ACPP/SP compared against ACPP/BP. Integer solutions were found for 4 out of 14 instances, spending less time than ACPP/BP. Note that computational savings range from 35.32% (D201_M) to 93.13% (D202_MTW). For two of these instances, the value of the linear relaxation matches the integer solution (D202_M and D300_M) whereas in the two remaining cases (D202_MT and D202_MTW), the number of required crews in the integer solutions was larger than the linear relaxation value shown in parenthesis. Additionally, non-integer solutions match ACPP/BP results for three instances, but in less computational time. Furthermore, note that the ACPP/SP provides tighter linear relaxations than those obtained with ACPP/BP. With the ACPP/SP model, it is possible even to solve the linear relaxation of larger instances, including a full-week instance (D202_MTWTFSS).

Table 2: Results for the ACPP/SP

Instance	Legs	ACPP/BP		ACPP/SP			Improvement CPU time
		Required crews	CPU time (s)	Required crews	CPU time (s)	Integer sol?	
D202_M	24	4	2.41	4	0.73	yes	69.49%
D202_MT	48	4	124.56	5 (4)	18.80	yes	84.91%
D202_MTW	72	4	3557.02	6 (4)	244.38	yes	93.13%
D202_MTWT	96	4	31427.63	4	2137.94	no	93.20%
D202_MTWTF	120	-	--	5	15392.20	no	-
D202_MTWTFS	138	-	--	5	96754.12	no	-
D202_MTWTFSS	152	-	--	6	122371.00	no	-
D201_M	99	16	89.36	16	57.80	no	35.32%
D201_MT	198	-	--	16	5473.89	no	-
D201_MTW	297	-	--	16	351045.23	no	-
D300_M	45	8	10.53	8	5.11	yes	51.49%
D300_MT	89	9	--	9	290.06	no	-
D300_MTW	134	-	--	9	7379.73	no	-
D300_MTWT	178	-	--	9	72692.00	no	-

One of the main reasons why computational time is so long is because decision variables in the subproblem are defined for all possible flight connections meeting precedence relations and time windows. However, we analyzed ACPP/SP solutions in depth and found no evidence of connections between flights more than two days apart (e.g., a leg on Monday does not connect with a leg on Thursday or a day after that). Taking into account the specific instances on hand we modified the mathematical formulation of the subproblem by pruning many variables, that is, we created only those variables allowing connections between legs no more than two days apart. Table 3 reflects the dramatic computational savings in MP(Ω_1) by applying this variable pruning strategy especially in large problems. For instance, the full-week instance D202_MTWTFSS was solved 48.40% faster compared to the original ACPP/SP formulation. Finally, note that this strategy is supported by the fact that the instances reflect the operational policies of the airline, but in general, this pruning strategy could sacrifice optimality or even worse, solution feasibility.

Table 3: Results for the ACPP/SP with variable pruning strategy

Instance	Legs	ACPP/SP original		ACPP/SP less variables			Improvement CPU time
		Required crews	CPU time (s)	Required crews	CPU time (s)	Integer sol?	
D202_M	24	4	0.73	4	0.36	yes	50.95%
D202_MT	48	5(4)	18.80	5 (4)	21.11	yes	-12.31%
D202_MTW	72	6(4)	244.38	6 (4)	255.16	yes	-4.41%
D202_MTWT	96	4	2137.94	4	1523.25	no	28.75%
D202_MTWTF	120	5	15392.20	5	10917.50	no	29.07%
D202_MTWTFS	138	5	96754.12	5	45825.80	no	52.64%
D202_MTWTFSS	152	6	122371.00	6	63137.40	no	48.40%
D201_M	99	16	57.80	16	23.50	no	59.34%
D201_MT	198	16	5473.89	16	4137.16	no	24.42%
D201_MTW	297	16	351045.23	16	160639.00	no	54.24%
D300_M	45	8	5.11	8	2.44	yes	52.29%
D300_MT	89	9	290.06	9	131.47	no	54.68%
D300_MTW	134	9	7379.73	9	3423.08	no	53.62%
D300_MTWT	178	9	72692.00	9	55903.30	no	23.10%

After analyzing several execution profiles we observed that the problem is highly degenerated. This can be partially explained because all pairing costs were equal (c^k = 1), thus causing no differentiation between pairings. Therefore, we defined a new column pricing criterion based on pairing sit time, that is, the time during which a crew is not doing activities such as flying or effectively serving on the ground, among other tasks. We chose this criterion to make the most efficient possible use of each crew, assuming that each minute a crew is idle, the company is likely to be wasting a very expensive resource. Therefore, crew cost (c^k) was replaced by a fixed crew cost (c^c) and a variable cost related with crew sit time (c^{cm}). Table 4 shows the results obtained with the new pricing criterion. The first important remark is an improvement in the number of required crews if the solution to the ACPP/SP is integer for 7 out of 8 instances (e.g., D202_MT) while maintaining the same values for the linear relaxation than with previous models. Moreover, integer solutions were obtained for more instances with the new pricing criterion than with the original implementation of the ACPP/SP and with the pruning strategy (e.g., D202_MTWT). However, these improvements are obtained under higher computational times.

Table 4: Results for the ACPP/SP using the sit time pricing criterion

Instance	Legs	ACPP/SP less variables			ACPP/SP sit time			Improvement CPU time
		Required crews	CPU time (s)	Integer sol?	Required crews	CPU time (s)	Integer sol?	
D202_M	24	4	0.36	yes	4	0.45	yes	-25.83%
D202_MT	48	5 (4)	21.11	yes	4	50.20	yes	-137.82%
D202_MTW	72	6 (4)	255.16	yes	5 (4)	244.45	yes	4.20%
D202_MTWT	96	4	1523.25	no	5 (4)	3026.88	yes	-98.71%
D202_MTWTF	120	5	10917.50	no	8(5)	45252.40	yes	-314.49%
D202_MTWTFS	138	5	45825.80	no	8(6)	92593.60	yes	-102.06%
D202_MTWTFSS	152	6	63137.40	no	--	--	--	--
D201_M	99	16	23.50	no	16	83.31	yes	-254.51%
D201_MT	198	16	4137.16	no	17(16)	48483.00	yes	-1071.89%
D201_MTW	297	16	160639.00	no	--	--	--	--
D300_M	45	8	2.44	yes	8	3.44	yes	-41.10%
D300_MT	89	9	131.47	no	9	572.47	yes	-335.44%
D300_MTW	134	9	3423.08	no	10(9)	11361.10	yes	-231.90%
D300_MTWT	178	9	55903.30	no	--	--	--	--

Finally, Table 5 shows the results for the ACPP/SC including deadheads and layovers. In this case, deadheads were penalized with a cost greater than that of a crew, assuming they are extremely undesirable (may change with the airline's policies). Based on the good results obtained with variable pruning and pairing pricing differentiation on the ACPP/SP, we kept these changes for the ACPP/SC. Note that integer solutions were found for 12 out of 14 instances and the value of the linear relaxation was improved. It is important to point out that a solution for a full-week instance (D202_MTWTFSS) was obtained in nearly 11 hours. Although this may seem as a long computational time, the airline plans its operations only four times per year: end-of-year holidays, Holy Week, summer vacation, and low season. Once planned, a weekly schedule is repeated throughout the season. In addition, it is important to compare this time with the three hours it takes the airline to manually plan much simpler daily activities.

Table 5: Results for the ACPP/SC

Instance	Legs	Required crews	CPU time (s)	Integer sol?
D202_M	24	4	0.56	yes
D202_MT	48	4	40.64	yes
D202_MTW	72	5(4)	419.09	yes
D202_MTWT	96	4	1928.91	yes
D202_MTWTF	120	6(5)	21367.80	yes
D202_MTWTFS	138	6	24424.30	yes
D202_MTWTFSS	152	7(6.5)	38356.50	yes
D201_M	99	16	76.70	yes
D201_MT	198	18(16)	40296.90	yes
D201_MTW	297	21 (16)	454811.00	yes
D300_M	45	8	1.19	yes
D300_MT	89	9	106.61	yes
D300_MTW	134	9	4079.89	yes
D300_MTWT	178	12(9)	7076.46	yes

8. Concluding Remarks and Future Research

We proposed a set of optimization models to minimize the number of crews required to cover all legs in a weekly flight schedule for a mid-sized commercial airline operating in the low-fare market. The ACPP/BP, which serves as a benchmark formulation, shows some computational limitations and fails to solve large instances. Modeling the ACPP as a SPP results in the ACPP/SP, a model that provides solutions that, although not always integer, can be used as a lower bound for other models or procedures. Allowing deadheads, relaxes the SPP into a SCP model, giving rise to the ACPP/SC. This model finds integer solutions for more (and larger) instances than the ACPP/BP and the ACPP/SP, and provides a wide variety of solutions for a particular instance depending on the airline's level of deadhead desirability. The proposed ACPP/SC formulation includes realistic concepts such as crew costs, sit-time costs, deadheads, and layovers.

With the ACPP/SC instances ranging from 24 legs (D202_M) to 297 legs (D201_MTW) were solved in computational times from 0.56 seconds to 11.20 hours. When compared to the current airline planning operations, it is worth mentioning that planning a single day takes 3 hours without any proof of the solution quality. On the other hand, with the proposed models the largest single-day instance runs in less than two minutes with a quality assurance. It is expected that the set of proposed models will improve crew planning in the airline so as to generate significant annual savings and to increase its competitiveness.

Future research currently underway includes the design and implementation of heuristic methods to improve the efficiency of the subproblem in the column generation technique. Particularly, a promising line of research is the extension of the results obtained with the heuristic based on *split* described in Vargas et al.

(2009) applied to the case of a multiple day planning horizon. It is also challenging and of practical interest to integrate models for the *Airline Crew Rostering Problem* with the ACPP so that it could become possible to define complete crew plans. To evaluate the models' scalability and performance it is important to try them on different airlines and with larger instances. Finally, the ACPP/SC formulation could be modified to consider flights that require two or more crews, such as it happens in overseas flights.

Acknowledgments

The authors would like to thank FICO for proving Xpress-MP Professional licenses under the academic partnership with Universidad de los Andes.

References

Ahmadbeygi, S., Cohn, A., and Weir, M. (2008). An integer programming approach to generating airline crew pairings. *Computers & Operations Research*, Volume 36, Issue 4, April 2009, pp. 1284-1298.   .

Ahmed, A.H. and Poojari, C.A. (2008). An overview of the issues in the airline industry and the role of optimization models and algorithms. *Journal of the Operational Research Society*. 59, pp. 267-277.

Anbil, R., Gelman, E., Patty, B., Tanga, R. (1991). Recent advances in crew-pairing optimization at American Airlines. *Interfaces*. Vol.21, No.1, January-February 1991, pp. 62-74.

Anbil, R., Forrest, J.J., Pulleyblank, W.R. (1998). Column generation and the airline crew pairing problem. *Documenta Mathematica*. Extra volume ICM III, pp. 677-686.

Arabeyre, J.P., Fearnley, J., Teather, W. (1969). The airline crew scheduling problem: a survey. *Transportation Science*. Vol.3. pp. 140-163.

Barnhart, C., Cohn, A.M., Johnson, E.L., Klabjan, D., Nemhauser, G.L., and Vance, P.H. (2003). Airline crew scheduling. In: *Handbook of Transportation Science* (R. Hall, ed.), Kluwer Academic Publishers, pp. 517-560.

Crawford, B., Castro, C., Monfroy, E. (2006). A hybrid ant algorithm for the airline crew pairing problem. *Lecture Notes in Computer Science* in *MICAI 2006*. Vol. 4293. Springer-Verlag Berlin Heidelberg. pp. 381-391.

Ching Chang, S. (2006). An improved genetic algorithm for the dynamic cargo crew pairing problem. (2006). *Proceedings of the 9th WSEAS International Conference on Applied Mathematics, Istanbul, Turkey, May 27-29, 2006*. pp. 205-213.

Ernst, A., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D. (2004) An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, pp. 21-144.

Gopalakrishnan, B., Johnson, E. (2005) Airline crew scheduling: state-of-the-art. *Annals of Operations Research*, 140(1), pp.305-337.

Guo, Y., Mellouli, T., Suhl, L., Thiel, M. (2006). A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. *European Journal of Operational Research*. Vol. 171. pp.1169-1181.

Hoffman, K.L., Padberg, M. (1993). Solving airline crew scheduling problems by branch-and-cut. *Management Science*. Vol. 39. Issue 6. pp.657-682.

Klabjan, D. (2005). Large-scale models in the airline industry. In: *Column generation* (M.M. Solomon, ed.), Springer, New York, pp. 163-195.

Lavoie, S., Minoux, M., Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*. Vol. 35. pp. 45-58.

Levine, D. (1996). Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations Research*. Vol. 23. pp. 547-558.

Marsten, R. E. and Shepardson, F. (1981) Exact solution of crew problems using the set partitioning mode, recent successful applications. *Networks*, 11, pp. 165-177.

Sylla, A. (2000). Operations research in the airline industry. Technical Report, University of Oulu, pp. 49-50. http://www.oulu.fi/seydou/aboud.pdf.

Pavlopoulou, C., Gionis, A. (1996). Crew pairing optimization based on CLP. University of Athens, Department of Informathics.

Vance, C., Atamtürk, A., Barnhart, C., Gelman, E., Johnson, E.L., Krishna, A., Mahidhara, D., Nemhauser. G.L., Rebello, R. (1997). Heuristic branch-and-price approach for the airline crew pairing problem. pp.29.

Vargas, M.A., Walteros, J.L., Medaglia, A.L. (2009). Hybrid approach to airline crew pairing optimization for small carriers. To appear in *Proceedings of the 2009 Industrial Engineering Research Conference*. Miami, FL. May 30 - June 3, 2009.

Yu, G. and Yang, J. (1998). Optimisation application in the airline industry. In: Du, D., Pardalos, P.M. (eds) *Handbook of Combinatorial Optimization*. Vol. 2, Kluwer Academic Publishers, Netherlands, pp. 635-726.