Les Cahiers du GERAD ISSN: 0711–2440

Simultaneous optimization of personalized integrated scheduling for pilots and copilots

A. Kasirzadeh, F. Soumis,
M. Saddoune, M. Towhidi

G–2014–111

December 2014

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2014.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2014.

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Simultaneous optimization of personalized integrated scheduling for pilots and copilots

Atoosa Kasirzadeh[a]
François Soumis[a]
Mohammed Saddoune[a,b]
Mehdi Towhidi[a]

[a] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

[b] *Faculté des Sciences et Techniques de Mohammedia, Université Hassan II, Mohammedia, Maroc*

atoosa.kasirzadeh@polymtl.ca
francois.soumis@gerad.ca
mohammed.saddoune@polymtl.ca
mehdi.towhidi@gerad.ca

December 2014

Les Cahiers du GERAD

G–2014–111

Copyright © 2014 GERAD

Abstract: The airline crew scheduling problem involves assigning a group of crew members to scheduled flights over a planning horizon (usually a month) while respecting safety rules and regulations. Because of its size and complexity, this problem is frequently solved in two steps, first crew pairing and then crew assignment. Therefore, the global optimization of the crew scheduling is not guaranteed, because the crew pairing problem does not take into account the scheduling constraints. The problem of integrated bidline scheduling (anonymous schedules) for pilots has been investigated by Saddoune et al. In this paper, we deal with the integrated personalized crew scheduling problem. In this case, personal preferences and constraints result in different monthly schedules for the pilots and copilots. However, to maintain the robustness of the crew schedules under perturbation at the operational level, the pilots and copilots must have similar pairings when possible. This paper presents a heuristic algorithm that alternates between the pilot and copilot scheduling problems to obtain similar pairings even when the monthly schedules are different. Each problem is formulated as a set partitioning problem, and the solution approach is based on column generation and constraint aggregation. We conduct computational experiments on a set of real instances from a major US carrier.

Key Words: Airline crew scheduling, integrated crew scheduling, integrated airline crew pairing and assignment, personalized crew scheduling, simultaneous pilot and copilot scheduling, large-scale scheduling, OR in airlines, column generation, dynamic constraint aggregation.

Acknowledgments: This research was supported by the Natural Sciences and Engineering Research Council of Canada and a collaborative R&D grant from AD OPT, a division of Kronos. We thank the personnel of AD OPT for providing the data sets and the GENCOL software library. We are also grateful to François Lessard for his advice.

1 Introduction

Operations Research (OR) approaches have contributed extensively to the tools and solution methodologies for the large-scale decision problems faced by airlines. In this context, OR approaches aim to reduce the cost of the airline operations and increase the quality of the crew schedules. Because of its complexity and the likelihood of disruption, the airline decision process is frequently divided into two related procedures: *planning* and *recovery*. The planning for a specific month is carried out four to six weeks in advance. However, in real life, perturbations may occur because of the weather conditions, aircraft maintenance issues, crew problems, and other unplanned events. These perturbations may lead to delayed or canceled flights, affect the crew schedules, and alter passenger itineraries. The recovery procedure is designed to handle these perturbations and to recover the flight and crew schedules. This procedure is not within the scope of this paper, and we do not discuss it further.

Ideally, we would formulate the planning problem as a single optimization problem in which a generic, unified objective function maximizes the total expected profit of the airline. This ideal optimization problem would encompass all the planning steps and all the constraints and rules. In practice, however, because of its complexity, the problem is simplified. Each step is considered individually, and the output of one step is the input for the next. The airline planning problem is usually divided into four main steps: *flight scheduling*, *fleet assignment*, *aircraft maintenance and routing*, and *crew scheduling*. These steps are explained in more detail in Kasirzadeh et al. (2014).

In the literature, the crew scheduling problem is divided into two substeps, because of its size and computational complexity: *crew pairing* and *crew assignment*. Crew pairing is the problem of constructing anonymous pairings such that the cost of the pairings is minimized and the scheduled flights are covered. A *pairing* is a sequence of duties (working days) and layovers (overnight stops) for an unspecified crew member; a pairing starts and ends at a base. In short- and medium-haul problems, pairings typically last one to five days; in long-haul problems, longer pairings are also allowed. Each crew member is associated with a *base* located at a large airport. The crew assignment problem is separable by crew category and aircraft type (or family of types). The crew categories are *cockpit crew members* and *cabin crew members*. The cockpit crew members are trained to fly one or more aircraft types. The cockpit crew always contains a pilot and a copilot, and for some large aircraft a flight engineer is added. The cabin crew members (the cabin captain and the flight attendants) are responsible for passenger services and safety, and they can be assigned to multiple aircraft types. Cockpit crew are paid significantly more than cabin crew, because of the expertise needed for their assigned tasks. As a result, most crew scheduling research has focused on cockpit scheduling (the flight engineers are not taken into account).

Crew scheduling is usually solved as either a *bidline* or *personalized* assignment problem. In the bidline approach, monthly crew schedules are constructed anonymously and then assigned to crew members. A *schedule (monthly schedule)* is a sequence of pairings separated by time off. The crew members bid for their preferred schedules. The personalized assignment problem takes into account vacations and training periods as well as the crew preferences. This problem is either treated as *rostering*, which maximizes the global satisfaction of the crew members, or *seniority-based* scheduling, which maximizes the satisfaction of the crew members in seniority order.

Solving the airline planning problem in multiple steps clearly does not give a fully optimal result. Recently, researchers have combined two or more of the steps in order to obtain better solutions.

In the context of integrated flight scheduling and fleet assignment, Lohatepanont and Barnhart (2004) introduce models and heuristic algorithms. They present experiments on medium-sized data from a major U.S. airline. Sherali et al. (2013) propose a Benders decomposition approach; they take flexible flight times, schedule balance concerns, and recapture issues into account. They provide results for small data sets from United Airlines.

In the context of integrated aircraft routing and crew pairing, Cordeau et al. (2001) describe a set covering model. They propose an algorithm based on Benders decomposition and column generation, and good results are reported for a three-day horizon. Cohn and Barnhart (2003) introduce heuristic and optimal algorithms;

they report experiments on two small instances. Mercier et al. (2005) present two Benders decomposition approaches: one considers routing to be the master problem and the other considers pairing to be the master problem. Their experiments with data from two major airlines give good solutions. Weide et al. (2010) introduce a procedure that iterates between routing and pairing; the solution of one step is used when the other step is being solved. This procedure tries to create robust solutions by reducing the number of crew and aircraft changes.

Sandhu and Klabjan (2007) present an integrated model for fleet assignment and crew pairing; they neglect the aircraft maintenance constraints. They present two algorithms: 1) Lagrangian relaxation combined with column generation and 2) Benders decomposition. Results are provided for small instances. Gao et al. (2009) describe a model and algorithm for robust integrated fleet and crew planning. Their experiments give good results for data from a major U.S airline.

Klabjan et al. (2002) develop a model and algorithm for the integration of flight scheduling, aircraft routing, and crew pairing. They present good results for small instances. For the same problem, Papadakos (2009) introduces a set covering model and solves it via enhanced Benders decomposition combined with column generation. He reports good results for small data sets. Cacchiani and Salazar-González (2013) propose a heuristic and an algorithm based on column generation. They present results for small instances with no overnight flights.

The integration of the crew pairing and crew assignment problems has been investigated by Zeghal and Minoux (2006), Guo et al. (2006), Souai and Teghem (2009), Saddoune et al. (2012), and Saddoune et al. (2011). Zeghal and Minoux (2006) propose an integer linear programming model using clique constraints for integrated pairing and bidline assignment. They assume that the duties can be generated a priori, and deadheads can be introduced whenever required without extra cost. They solve the problem by two branch and bound algorithms, one exact and one heuristic, and present good results for small instances for a five-day horizon (with up to 101 flights and 40 crew) and a horizon of one month (with up to 195 flights and 18 crew). Guo et al. (2006) introduce a partially integrated crew scheduling approach based on pairing-chain generation. For each base, given the total number of crew members stationed at that base, they construct a series of *pairing chains* containing weekly rests and then adjust these pairings to take into account the crew requests and prescheduled activities. The tests are conducted for a 15-day horizon (with up to 1977 flights and 188 crew) and a 31-day horizon (with 808 flights and 44 crew). Souai and Teghem (2009) describe a genetic algorithm for integrated pairing and personalized assignment. They provide results for three small instances and a monthly planning horizon (with up to 1872 flights and 68 pilots). Saddoune et al. (2012) develop a model and algorithm based on column generation and dynamic constraint aggregation for integrated pairing and bidline assignment. They report good results for seven data sets from a major North American airline; the largest has 7765 scheduled flights and 305 pilots. They report an average cost reduction of 3.37%, but the computational time was 6.8 times higher than that of the sequential approach. Saddoune et al. (2011) introduce different neighborhood strategies to reduce the size of the subproblems. The computational times are reduced by an average factor of 2.3 and the cost saving is 4.02% to 4.76%.

We propose to solve a set partitioning formulation that constructs monthly personalized schedules for cockpit crew members. Personalized scheduling is increasingly accepted at North American airlines because it offers several advantages over bidline scheduling. Personalized scheduling considers the employee's requests during the construction of the schedule and takes into account predefined employee activities (e.g., vacations, training). Furthermore, personalized schedules decrease the number of schedule adjustments at the operational level and increase productivity.

We consider four objectives: (i) minimizing the cost of constructing pairings, (ii) minimizing the cost of constructing monthly schedules, (iii) maximizing the global satisfaction of crew members, and (iv) maximizing the number of common duties and pairings for pilots and copilots. Objectives (i), (ii), and (iii) are obvious goals. In the sequential context, the solution of the pairing problem considers only objective (i). The bidline assignment problem takes into account only objective (ii). The personalized assignment problem considers objectives (ii) and (iii). Both the bidline and personalized assignment problems satisfy objective (iv) because the pairings are not changed as the schedules are constructed, and they are the same for pilots and copilots. Integrated bidline scheduling considers objectives (i) and (ii) and satisfies (iv) because the schedules are the same for pilots and copilots. Integrated personalized scheduling takes into account objectives (i), (ii), and (iii). However, for objective (iv), it is important to solve the pilot and copilot scheduling problems simultaneously, so that the pilots and copilots have similar pairings whenever possible.

We optimize the pilots and copilots simultaneously, taking their preferences into account. To the best of our knowledge, this paper is the first to consider the simultaneous optimization of cockpit crew scheduling where the pairings and personalized monthly schedules are constructed by an integrated approach. We solve the problem using a recent mathematical programming approach based on column generation (CG), called dynamic constraint aggregation (DCA). The general structure of the problem is presented in Figure 1.

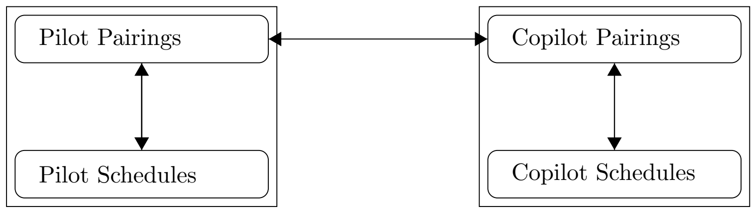


Figure 1: Schematic of simultaneous personalized integrated scheduling for pilots and copilots

The remainder of this paper is structured as follows. Section 2 provides a detailed description of our problem and the proposed heuristic. Section 3 describes the mathematical formulation, and Section 4 describes our algorithm in detail. Section 5 presents results for three data sets, and Section 6 provides concluding remarks.

2 Problem statement

The problem of constructing monthly schedules for airline crew members is different at different airlines, because the safety rules and regulations, preassigned activities, and crew preferences vary. In airlines where crew preferences are taken into account, there must be a trade-off between minimizing the cost of the crew schedules and satisfying the preferences of the crew members.

The cockpit crew scheduling problem is often solved under the assumption that the pilots and copilots are identical. In the context of bidline scheduling, this assumption is reasonable because the schedules are constructed anonymously and the crew preferences and preassigned activities are not taken into account. However, in the context of personalized scheduling, the pilots and copilots are not identical because they have different personal preferences. In this case, the *simultaneous optimization* of cockpit crew scheduling becomes relevant. We simultaneously construct monthly schedules for cockpit crew members taking into account the four objectives listed in Section 1. For more robust schedules, it is necessary to keep a pilot-copilot pair together during duties and pairings whenever possible. There are four reasons for this. First, it helps to ensure the robustness of duties. If the pilot and copilot are separated when a perturbation occurs, the propagation effect will be greater. Second, it helps to ensure the robustness of the rest periods between the duties. Third, it helps to reduce the number of briefings and debriefings. Finally, it helps to reduce the costs of hotels and taxis when perturbations occur. We develop an algorithm that aims to build common pairings and duties for pilots and copilots, minimize the cost of the pairings, and provide a global level of preference satisfaction. The details of our algorithm are explained below.

We propose a heuristic approach that iterates between the *pilot scheduling problem (PSP)* and the *copilot scheduling problem (CSP)*. It is outlined in Figure 2.

At each iteration of the algorithm, the personalized integrated scheduling problem is solved using the specialized DCA algorithm for the airline crew scheduling problem. The approach was developed by Elhallaoui et al. (2010) and specialized for airline crew scheduling problems by Saddoune et al. (2012). It is explained in Section 4. The objective of PSP (CSP) is to minimize the total cost of the monthly schedules for pilots (copilots) and to satisfy a global level of crew preferences. The two preferences that we take into account are

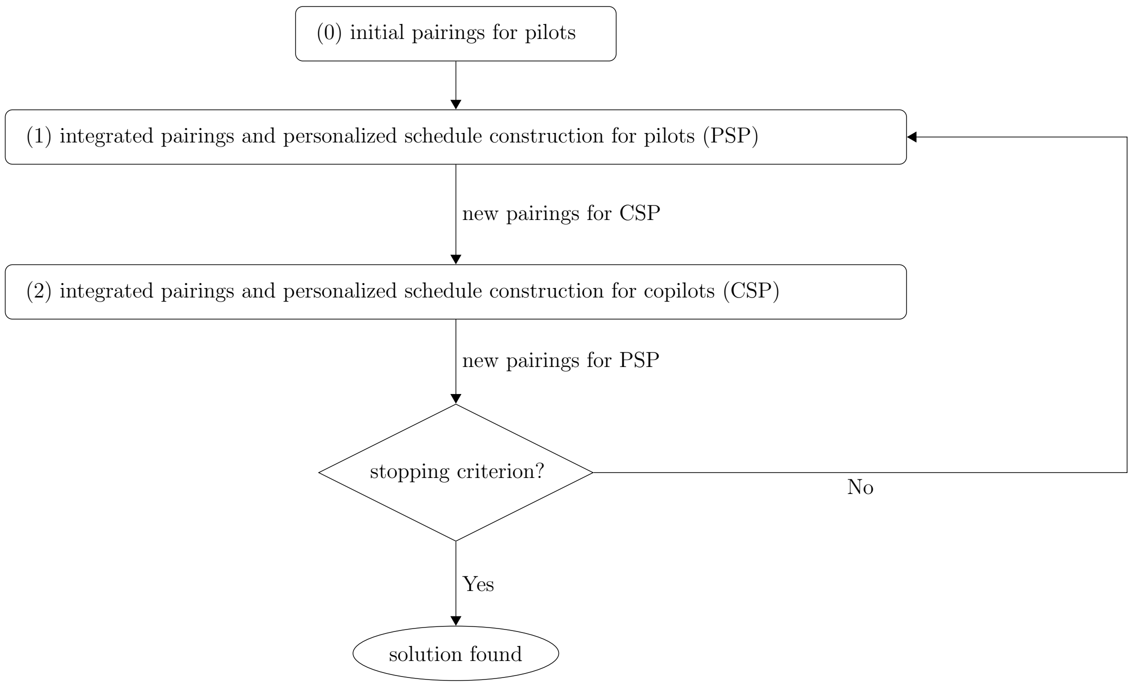


Figure 2: Heuristic algorithm flowchart for integrated crew scheduling problem

preferred vacations and *preferred scheduled flights*. These preferences are generated by two random generators developed by Kasirzadeh et al. (2014). The heuristic algorithm stops when a stopping criterion is satisfied. It starts from a set of initial pairings (phase 0). For this set we use the anonymous pairings constructed by Saddoune et al. (2013) for the same set of scheduled flights and safety regulations. In the first phase we solve PSP and construct personalized monthly schedules for the pilots. The set of initial pairings is updated accordingly, and the new set of pairings is called NPP (new pairings for pilots). Taking NPP into account, in the second phase we solve CSP and construct personalized monthly schedules for the copilots. We call the new set of pairings NPC (new pairings for copilots). Given NPC, we solve PSP again to obtain an updated NPP. We then solve CSP again. This process continues until a stopping criterion (a maximum number of iterations) is satisfied. We need this because it may take a long time for the algorithm to converge. There are two ways to force common sets of duties for the pilots and copilots. One option is to introduce soft constraints: bonuses for the common duties and pairings. The other is to introduce hard constraints; this restricts the domain of exploration and thus reduces the total computational time. We use hard constraints; these are explained in more detail in Section 4. Both the PSP and the CSP are mathematically formulated as set partitioning problems, as discussed in Section 3.

3 Mathematical formulation

The integrated personalized cockpit crew scheduling problem is mathematically formulated using the following notation:

Sets

F: set of scheduled flights;

P: set of feasible pairings;

L: set of pilots;

O: set of copilots;

V_l: set of vacation preferences for pilot $l \in L$;

V_o: set of vacation preferences for copilot $o \in O$;

G_l: set of preferred flights for pilot $l \in L$;

G_o: set of preferred flights for copilot $o \in O$;

S_l : set of feasible schedules for pilot $l \in L$;

S_o: set of feasible schedules for copilot $o \in O$;

Parameters

C_p: cost of feasible pairing $p \in P$;

C^l_s: cost of personalized schedule $s \in S_l$ for pilot $l \in L$;

C^o_s: cost of personalized schedule $s \in S_o$ for copilot $o \in O$;

$\bar{C}_f$: penalty for not covering flight $f \in F$;

n^l_s: number of preferred flights in schedule $s \in S_l$ for pilot $l \in L$;

n^o_s: number of preferred flights in schedule $s \in S_o$ for copilot $o \in O$;

c^l_f: bonus for covering preferred flight $f \in G_l$ for pilot $l \in L$;

c^o_f: bonus for covering preferred flight $f \in G_o$ for copilot $o \in O$;

c^l_v: penalty for not covering vacation preference $v \in V_l$;

c^o_v: penalty for not covering vacation preference $v \in V_O$;

$$e^{s,l}_f = \begin{cases} 1 & \text{if flight } f \in F \text{ is covered by pilot } l \in L \text{ in schedule } s \in S_l \\ 0 & \text{otherwise;} \end{cases}$$

$$e^{s,l}_p = \begin{cases} 1 & \text{if pairing } p \in P \text{ is covered by pilot } l \in L \text{ in schedule } s \in S_l \\ 0 & \text{otherwise;} \end{cases}$$

$$e^{s,o}_f = \begin{cases} 1 & \text{if flight } f \in F \text{ is covered by copilot } o \in O \text{ in schedule } s \in S_o \\ 0 & \text{otherwise;} \end{cases}$$

$$e^{s,o}_p = \begin{cases} 1 & \text{if pairing } p \in P \text{ is covered by copilot } o \in O \text{ in schedule } s \in S_o \\ 0 & \text{otherwise;} \end{cases}$$

$$v^{s,l}_v = \begin{cases} 1 & \text{if vacation } v \in V_l \text{ is covered by schedule } s \in S_l \\ 0 & \text{otherwise;} \end{cases}$$

$$v^{s,o}_v = \begin{cases} 1 & \text{if vacation } v \in V_o \text{ is covered by schedule } s \in S_o \\ 0 & \text{otherwise;} \end{cases}$$

Variables

$$x^s_l = \begin{cases} 1 & \text{if schedule } s \in S_l \text{ for pilot } l \in L \text{ is chosen} \\ 0 & \text{otherwise;} \end{cases}$$

$$x^s_o = \begin{cases} 1 & \text{if schedule } s \in S_o \text{ for copilot } o \in O \text{ is chosen} \\ 0 & \text{otherwise;} \end{cases}$$

$$\bar{e}_f = \begin{cases} 1 & \text{if flight } f \in F \text{ is not covered} \\ 0 & \text{otherwise.} \end{cases}$$

At each iteration of the algorithm, we solve either the PSP (1)–(4) or the CSP (5)–(8).

$$\min \sum_{l\in L}\sum_{s\in S_l} C_s^l x_l^s + \sum_{f\in F} \bar{e}_f \bar{C}_f \quad (1)$$

$$\text{s.t.} \sum_{l\in L}\sum_{s\in S_l} e_f^{s,l} x_l^s + \bar{e}_f = 1, \quad \forall f \in F \quad (2)$$

$$\sum_{s\in S_l} x_l^s \leq 1, \quad \forall l \in L \quad (3)$$

$$x_l^s \in \{0,1\}, \quad \forall l \in L, \forall s \in S_l \quad (4)$$

$$\min \sum_{o\in O}\sum_{s\in S_o} C_s^o x_o^s + \sum_{f\in F} \bar{e}_f \bar{C}_f \quad (5)$$

$$\text{s.t.} \sum_{o\in O}\sum_{s\in S_o} e_f^{s,o} x_o^s + \bar{e}_f = 1, \quad \forall f \in F \quad (6)$$

$$\sum_{s\in S_o} x_o^s \leq 1, \quad \forall o \in O \quad (7)$$

$$x_o^s \in \{0,1\}, \quad \forall o \in O, \forall s \in S_o \quad (8)$$

Objective (1) (objective (5)) minimizes the total cost associated with the pilot (copilot) schedules. Constraints (2) (constraints (6)) ensure that each scheduled flight is assigned to exactly one pilot (copilot). Constraints (3) (constraints (7)) assign at most one schedule to each pilot (copilot), and constraints (4) and (8) are the binary requirements for the variables.

The cost of a schedule is composed of the cost of the pairings and the bonuses and penalties for preferences. In practice, the cost of a pairing has a complex nonlinear structure and an approximate cost function is often used. The pairing cost function that we use was introduced by Mercier et al. (2005) and enhanced by Saddoune et al. (2013). It considers waiting, deadheading, and the duty cost to be the elements of a pairing cost. The preferences are translated into bonuses and penalties and are also considered. The cost of personalized schedule s for pilot $l \in L$ is

$$C_s^l = \sum_{p\in P} e_p^{s,l} C_p + n_s^l c_f^l + \sum_{v\in V_l} (1 - v_v^{s,l}) c_v^l.$$

The cost for copilot $o \in O$ is simply obtained by substituting o for l. Because preferences vary from one crew member to another, we have one subproblem for each crew member, and so personalized scheduling is more complex than bidline scheduling.

4 Algorithm

Our algorithm is an enhanced version of CG embedded within branch and bound. CG is an iterative approach used to solve the relaxation of large-scale linear programming (LP) problems with set partitioning constraints (Desrosiers and Lübbecke (2005), Desrosiers et al. (1995), Barnhart et al. (1996)). Because of degeneracy, CG becomes inefficient when the number of set partitioning constraints is large and the columns are dense (with on average more than 8–12 nonzero elements). In our problem, the number of nonzeros varies between 30 and 45. If more than 90% of the constraints of a large-scale LP are set partitioning constraints, DCA can be combined with CG to reduce degeneracy and accelerate the solution process.

DCA aggregates some of the set partitioning constraints at each iteration of the restricted master problem (RMP). The theoretical framework for DCA was introduced by Villeneuve (1999), and Elhallaoui et al. (2005) provided the first implementation. An *equivalence relation* is defined for the set C of columns with positive values in the initial solution: two tasks t_1 and t_2 are *equivalent* if every column in C covers both t_1 and t_2 or neither. We use this relation to form the tasks into *clusters*. DCA starts with a feasible or infeasible initial solution (obtained by a heuristic, by logical reasoning, or from a previous solution) and corresponding set C. The set C is modified, when necessary, until an optimal solution is found. At each iteration, DCA changes the RMP to an aggregated restricted master problem (ARMP), which is smaller and easier to solve. We solve the ARMP by an LP optimizer and compute a pair of primal-dual solutions for the aggregated constraints. To generate columns for the original problem, we need a dual disaggregation process to provide dual solutions for each constraint of the RMP. We perform this disaggregation based on shortest-path calculations to provide a value for each set partitioning constraint of the original problem.

A *compatibility* criterion is defined between a partition Q and the path variables. A path said to be compatible with a partition if, for each cluster of the partition, it covers either all of the cluster's tasks or none. A newly generated column can be added to ARMP if it is compatible with the current partition. Otherwise, the variable is incompatible, and it can be added to ARMP only if the partition is modified; this is because ARMP contains only compatible columns. The criterion for updating the partition is based on the relationship between the reduced cost of the least compatible column reduced cost (CCR) and the least incompatible column reduced cost (ICR). This relationship is such that CCR is less than ICR times a predetermined multiplier Γ ($\Gamma = 1$ for our tests).

To benefit from the current partition and the compatible variables, Elhallaoui et al. (2010) described a version of DCA with multiple phases (MPDCA) that favors the generation of compatible or slightly incompatible columns with respect to the current partition; it uses a partial pricing strategy that favors slow disaggregation. To apply this strategy, we define a phase number (h) and an incompatibility number (r) for each column such that at phase h only the incompatible variables with $r \leq h$ are priced out. The value of r is an approximation of the number of additional clusters needed to make an incompatible column compatible. At the beginning of the solution process, we set the phase number to 0. We solve the ARMP and calculate the optimal primal and disaggregated dual solutions and the incompatibility number for each column. We then apply the partial pricing strategy and price out the columns with $r = 0$ (the compatible columns). When we find no negative-reduced-cost column, we increment h and go to the next phase ($h + 1$) or stop if h is the final step. If negative-reduced-cost columns are found, we determine whether or not the current partition must be modified. If no change is required, we add all or a subset of the compatible columns to the ARMP, and the MPDCA moves to its next iteration. Otherwise, we update the partition.

In the algorithm of Saddoune et al. (2012) for the integrated bidline pilot scheduling problem, a cluster is a pairing, and a set of pairings that covers all the scheduled flights is a partition. Each column corresponds to a feasible monthly schedule for a pilot (or copilot). We use the specialized MPDCA algorithm of Saddoune et al. (2012) to solve the integrated personalized crew scheduling problem. Our initial set of pairings is the solution of Saddoune et al. (2013). MPDCA is exact if the last phase number k is sufficiently large to ensure the pricing of all feasible columns. Given the complexity of the problem, we use only k=0 and k=1 for our experiments. In other words, our MPDCA is heuristic. In practice, to avoid the well-known tailing-off effect, CG is stopped before optimality is reached. Two parameters determine the CG stopping criterion. We stop the CG if in the last i iterations the objective value has decreased by less than a threshold value. These values are chosen based on preliminary tests: we stop the CG if within the previous 25 iterations, the objective value has decreased by less than 0.01%. This greatly reduces the search domain for the optimization. The new method starts with the sequential solution and improves it, and when we choose the above parameters we must find a trade-off between computational time and solution quality. Our results show that the sequential solution is substantially improved.

At each node of the branch and price tree, CG seeks a near-optimal linear relaxation solution. Two branching strategies are considered. The first fixes to 1 all the fractional values that are greater than a predetermined threshold (0.85 for our tests). The second forces two flights to be consecutive in a pairing. We choose the branching strategy for a given node by computing a score for each strategy and choosing the strategy with the higher score. Saddoune et al. (2012) showed that compared to the sequential approach, integrated bidline pilot scheduling reduces the cost by reducing the number of pilots and finding better schedules. Recall that our objective function is difficult. It minimizes the cost of the schedules, maximizes the satisfaction of the preferences, and encourages common pairings for pilots and copilots.

We can associate an acyclic directed time-space network $G = (N, A)$, where N and A are the node and arc sets, with each subproblem (i.e., each employee) of the PSP (and CSP). This network has five node types: source, sink, opportunity, departure, and arrival. It has twelve arc types: start of schedule, end of schedule, flight, deadhead, preferred vacation, rest, wait time, start of duty, start of pairing, day off, post-pairing, post-pairing rest. It is similar to the subproblem network in Saddoune et al. (2012). The difference is that in Saddoune et al. (2012) one subproblem is associated with each base, so there are just three subproblems. We have one subproblem for each employee (pilot or copilot). Furthermore, we have an additional arc type for preferred vacations.

To solve the subproblems we must find columns (shortest paths) with negative reduced costs that satisfy the resource constraints. Several local constraints and regulations are modeled by resources. A *resource* is a quantity that varies along a path. Each resource is distinguished by two characteristics: the *resource window* and the *resource consumption*. A resource window is associated with each node of the network. We build partial paths starting from the source node, and resource consumption occurs when an arc is added to the partial path according to the resource extension functions. We can add an arc only if the resource consumption of the new path is within the resource window of the new node. There are nine resources: maximum number of landings in a duty, maximum number of duties in a pairing, maximum working time in a duty, minimum working time in a duty, maximum duration of a duty, maximum pairing duration, minimum number of days off in a schedule, maximum number of consecutive working days, and maximum credited flying time. A feasible source-to-sink path (crew schedule) in a subproblem is a path that satisfies the resource constraints. For our subproblems, we use the resource values of Saddoune et al. (2012).

The subproblems are solved using dynamic programming. For resource-constraint networks, a multidimensional labeling algorithm is used (Irnich and Desaulniers, 2005). At the end node of each partial path, we consider a label with multiple elements, an element for each resource value and an element for the reduced cost. The labels are set to 0 at the source node. They are modified by label extension functions when an arc is added to a partial path. For large subproblems, many labels can be generated. To reduce the time for the path generation, we apply a *label dominance* rule that removes some partial paths. A partial path with end-node label L_1 is dominated by label L_2 if each component of L_1 is less than or equal to the corresponding component of L_2 (the value of at least one component should be strictly less).

We use a heuristic and an exact version of the label-setting algorithm of Saddoune et al. (2012) to solve the linear relaxations. At each LP iteration, we first use a heuristic in which the dominance rule considers the reduced cost along with a subset of the resources. This heuristic is relatively fast, but there is no guarantee that it finds the shortest path. If no negative-reduced-cost paths are found, an exact version uses all the resource components in the labeling algorithm. Based on the preliminary computational results, the following five resources are considered for the heuristic: maximum duration of a duty, maximum working time in a duty, minimum working time in a duty, maximum pairing duration, and maximum number of consecutive working days.

5 Computational results

In this section, we present results for the integrated personalized cockpit crew scheduling problem for monthly flight schedules operated by short-haul aircraft. Three data sets are provided by a major North American airline; they are described in Table 1. The initial pairings that we use in step (0) of the algorithm are the results of Saddoune et al. (2013), where the pairing problem is solved for the same data set. The number of pilots (and copilots) for each instance is the number of pilots in the solution of Saddoune et al. (2012) for the sequential bidline scheduling problem. We consider two types of preferences: vacations and preferred flights. The preferences are generated using the random generators developed by Kasirzadeh et al. (2014). The heuristic algorithm is implemented in C++, and version 4.5 of GENCOL is used. The RMPs are solved using CPLEX 12.4. We performed our tests on a computer equipped with an Intel(R) Core(TM) processor clocked at 3.40 GHz.

Table 1: Characteristics of instances

	$I1-727$	$I2-DC9$	$I3-D94$
No. of Scheduled Flights	1013	1500	1854
No. of Pilots (Copilots)	33	34	47
No. of Stations	26	35	41
No. of Initial Pairings	172	303	274

All the instances have three crew bases. For the feasibility of the pairings, we use the parameter values of Mercier et al. (2005), and for the feasibility of the schedules, we use the parameter values of Saddoune et al. (2012). In addition, we define some new parameters including a penalty for failing to cover a scheduled flight, a bonus for covering preferred flights, and a penalty for failing to satisfy a vacation preference. The costs of failing to cover scheduled flights, failing to satisfy vacation preferences, covering preferred flights, and covering scheduled flights are all related. We assume that it is more important to cover the scheduled flights than to satisfy the vacation and flight preferences. However, a very small percentage of uncovered flights is acceptable since most airlines have reserved cockpit crew members who can be allocated to uncovered flights. We assume that the cost of covering a scheduled flight is 0, and we set the cost of failing to cover a scheduled flight to 10000. Our results show that this cost is large enough to ensure that the percentage of uncovered flights is very small. Our preliminary results suggest a penalty of 5000 for failing to satisfy a vacation preference and a bonus of −50 for covering a preferred flight. These values ensure that a good percentage of the preferences are satisfied while the gap stays small and the percentage of uncovered flights remains very small.

In the first group of tests we consider variations of the penalty for failing to cover vacation preferences. The bonus for preferred flights is set to −50 and the vacation penalty varies between 1000 and 6000. The results for the three instances are given in Tables 2, 3, and 4. In each table, the first two rows give the penalty and bonus values. The *number of differing pairings* indicates the number of pairings that are not common to the pilots and copilots. The *pairing similarity* is the percentage of common pairings at the last iteration. The *number of differing duties* is the number of duties that are not common to the pilots and copilots. The *duty similarity* is the percentage of common duties at the last iteration. The *total no. of CG iterations* is the total number of CG iterations in the three iterations. The *total CPU time (in minutes)* is the total CPU time of the three iterations. The *gap* is the percentage difference between the lower bounds (LP solutions) and the upper bounds (integer solutions). The *percentage of uncovered flights* is the percentage of scheduled flights that remain uncovered at the end of the process despite the penalty.

To evaluate the quality of the solution, we use two indicators: the *percentage of preferred flights* and the *percentage of satisfied vacation preferences*. Increasing the penalty for failing to satisfy vacation preferences increases the percentage of satisfied preferences. However, a trade-off occurs when we allow a very small percentage of uncovered flights. We did not increase the vacation penalty beyond 6000 because either the gaps or the percentage of uncovered flights increased when we increased it from 5000 to 6000. The final solutions are found in a reasonable time. In practice, if the integrality gap is greater than 1%, it is advisable to use the solution of the penultimate iteration or to apply different branching strategies.

In the second group of tests we fix the vacation penalty at 5000, and the bonus for preferred flights varies between −10 and −60. The results for the three instances are given in Tables 5, 6, and 7; the information presented is the same as that in the earlier tables. We do not consider very low bonuses for covering preferred flights simply because the bonus is expressed as a negative cost, and this cost interacts with the real cost of the pairings. Preliminary results show that very low negative costs can result in a high percentage of uncovered flights. The results show that the algorithm is not very sensitive to the bonus for preferred flights. As mentioned, we did not have access to real data on employees' preferences, and these preferences are created

Table 2: $I1-727$ – Variations of vacation penalty

	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	1000		2000		3000		4000		5000		6000	
Preferred flight bonus	-50		-50		-50		-50		-50		-50	
No. of differing pairings	3		2		4		2		2		2	
Pairing similarity (%)	98.26		98.84		97.67		98.84		98.84		98.84	
No. of differing duties	3		2		4		2		2		2	
Duty similarity (%)	99.44		99.63		99.26		99.63		99.63		99.63	
Total no. of CG iterations	1103	969	815	925	740	1637	644	985	830	721	1066	701
Total CPU time (min)	3.75	3.21	2.76	3.02	2.38	1.38	2.14	3.22	2.93	2.35	3.54	2.24
Gap (%)	0.01	0.01	0.00	0.00	0.00	1.72	0.00	0.01	0.01	0.00	0.73	0.01
Uncovered flights (%)	0.00	0.00	0.00	0.00	0.99	0.99	0.00	0.00	0.00	0.00	1.18	1.18
Preferred flights (%)	24.68	24.38	26.65	24.68	26.75	23.49	26.46	23.69	26.55	25.07	26.67	24.58
Satisfied vacation preferences (%)	20.00	60.00	20.00	50.00	70.00	70.00	60.00	70.00	60.00	80.00	70.00	90.00

Table 3: $I2 - DC9$ – Variations of vacation penalty

	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	1000		2000		3000		4000		5000		6000	
Preferred flight bonus	-50		-50		-50		-50		-50		-50	
No. of differing pairings	0		1		0		0		1		0	
Pairing similarity (%)	100		99.67		100		100		99.67		100	
No. of differing duties	0		1		0		0		1		0	
Duty similarity (%)	100		99.86		100		100		99.86		100	
Total no. of CG iterations	2622	2527	2520	2670	2678	2621	2476	2273	2688	2538	2058	2786
Total CPU time (min)	14.06	14.45	15.20	15.11	16.49	9.24	15.20	12.60	16.77	14.51	12.24	16.37
Gap (%)	0.06	0.11	0.02	0.02	1.17	1.02	0.43	0.03	0.03	0.84	0.31	0.85
Uncovered flights (%)	0.00	0.40	0.00	0.00	0.40	0.53	0.93	0.93	0.00	0.40	0.40	0.53
Preferred flights (%)	26.53	26.13	26.73	26.60	26.71	26.41	26.11	25.84	26.00	26.07	26.40	25.77
Satisfied vacation preferences (%)	72.73	60.00	81.82	63.64	81.82	63.64	90.91	90.91	90.91	90.91	90.91	90.91

Table 4: $I3 - D94$ – Variations of vacation penalty

	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	1000		2000		3000		4000		5000		6000	
Preferred flight bonus	-50		-50		-50		-50		-50		-50	
No. of differing pairings	3		2		2		2		3		4	
Pairing similarity (%)	98.91		99.27		99.27		99.27		98.91		98.54	
No. of differing duties	4		2		2		3		4		4	
Duty similarity (%)	99.46		99.73		99.73		99.60		99.46		99.46	
Total no. of CG iterations	2226	2530	2414	2490	2918	2838	3018	2430	2107	2417	2916	2672
Total CPU time (min)	43.91	54.55	48.33	51.14	60.77	63.90	67.53	51.32	38.71	50.37	66.16	58.01
Gap (%)	0.04	0.02	0.74	0.44	0.39	0.03	0.13	1.01	0.01	0.52	1.01	0.03
Uncovered flights (%)	0.22	0.22	0.81	0.81	1.51	1.51	0.59	0.92	0.70	0.70	1.02	1.02
Preferred flights (%)	25.78	25.78	26.16	25.94	26.07	25.52	24.36	22.97	25.96	26.89	24.74	26.32
Satisfied vacation preferences (%)	45.45	53.33	60.00	60.00	80.00	73.33	86.67	73.33	86.67	80.00	86.67	73.33

Table 5: $I1 - 727$ – Variations of preferred flight bonus

	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	5000		5000		5000		5000		5000		5000	
Preferred flight bonus	-10		-20		-30		-40		-50		-60	
No. of differing pairings	2		3		2		2		2		2	
Pairing similarity (%)	98.84		98.26		98.84		98.84		98.84		98.84	
No. of differing duties	2		3		2		2		2		2	
Duty similarity (%)	99.63		99.44		99.63		99.63		99.63		99.63	
Total no. of CG iterations	832	550	754	1283	641	1056	596	738	830	721	474	834
Total CPU time (min)	2.76	1.82	2.52	4.28	2.15	3.59	2.09	2.46	2.93	2.35	1.73	2.79
Gap (%)	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.03
Uncovered flights (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Preferred flights (%)	26.60	24.58	26.65	24.58	26.75	24.58	26.46	24.68	26.55	25.07	26.75	24.98
Satisfied vacation preferences (%)	60.00	80.00	60.00	80.00	60.00	80.00	60.00	80.00	60.00	80.00	60.00	80.00

Table 6: $I2 - DC9$ – Variations of preferred flight bonus

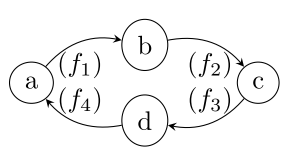
	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	5000		5000		5000		5000		5000		5000	
Preferred flight bonus	-10		-20		-30		-40		-50		-60	
No. of differing pairings	0		2		2		1		1		2	
Pairing similarity (%)	100		99.34		99.34		99.67		99.67		99.34	
No. of differing duties	0		3		3		1		1		3	
Duty similarity (%)	100		99.59		99.59		99.86		99.86		99.59	
Total no. of CG iterations	2691	2774	2963	2650	2749	2551	2322	2836	2688	2538	2261	2373
Total CPU time (min)	16.20	16.26	18.59	15.58	17.38	14.89	14.13	17.13	16.77	14.51	13.12	13.80
Gap (%)	0.02	0.02	0.01	0.01	0.02	0.08	1.11	0.07	0.03	0.84	0.00	0.01
Uncovered flights (%)	1.13	1.13	1.07	1.07	0.13	0.13	0.27	0.27	0.00	0.40	1.80	1.80
Preferred flights (%)	26.84	24.81	26.28	26.08	26.10	25.17	26.54	25.07	26.00	26.07	26.82	26.41
Satisfied vacation preferences (%)	90.91	90.91	90.91	90.91	90.91	90.91	81.82	90.91	90.91	90.91	90.91	90.91

Table 7: $I3 - D94$- Variations of preferred flight bonus

	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots	Pilots	Co-pilots
Vacation penalty	5000		5000		5000		5000		5000		5000	
Preferred flight bonus	-10		-20		-30		-40		-50		-60	
No. of differing pairings	4		6		5		3		3		3	
Pairing similarity (%)	98.54		97.81		98.18		98.91		98.91		98.91	
No. of differing duties	4		8		6		3		4		3	
Duty similarity (%)	99.46		98.93		99.20		99.60		99.46		99.60	
Total no. of CG iterations	3259	2726	3108	2976	2185	2602	3049	3138	2107	2417	2374	2683
Total CPU time (min)	73.14	62.54	68.06	48.29	46.63	55.86	64.38	69.63	38.71	50.37	46.93	56.64
Gap (%)	0.08	0.21	0.91	0.47	0.01	0.01	0.08	0.07	0.01	0.52	0.52	2.16
Uncovered flights (%)	0.43	0.49	0.86	0.86	0.97	0.97	0.92	0.92	0.70	0.70	1.51	1.51
Preferred flights (%)	24.92	26.07	24.97	25.24	26.42	25.87	25.97	25.97	25.96	26.89	25.85	25.41
Satisfied vacation preferences (%)	86.67	73.33	86.67	73.33	86.67	73.33	86.67	73.33	86.67	80.00	86.67	73.33

by the random generators explained in detail in Kasirzadeh et al. (2014); the generators do not take into account correlations between the choices of preferred flights. It is difficult to determine the likely correlations, as the following example shows.

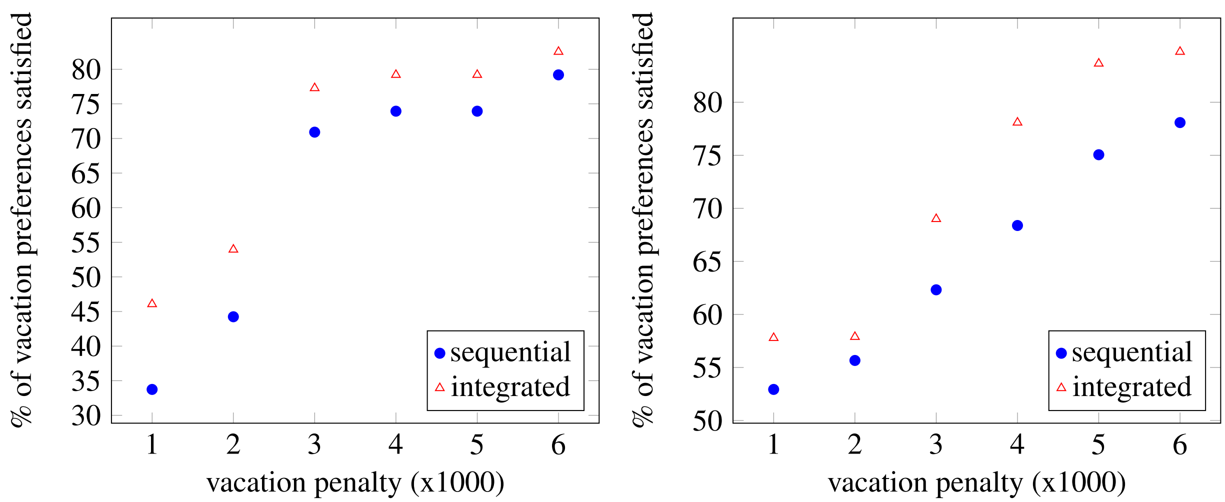
Consider a pairing composed of one duty that includes four flights between airports (a), (b), (c), and (d). If the pilot prefers (f_2), he/she may also choose as preferred flights (f_1) and a return path to base (a) that contains (f_3) and (f_4). He/she can also choose another combination of flights that can be included in a good pairing. These choices make it possible to assign the four preferred flights in a duty of four flights. However, we generate each employee's preferred flights by randomly choosing 10% of the flights from the set of pairings associated with his/her base. Therefore, it is difficult to have more than one or two preferred flights from a pairing. On average, there are five to seven flights per pairing, so the percentage of preferred flights is unlikely to be above 25%.



To improve the percentage of preferred flights, we need access to an expert who knows aircraft routing, and we do not have such access. In addition, we use a stopping criterion to reduce the computational time. The search for the optimal solution therefore does not explore all the possibilities.

We did not explore scheduling for medium-haul aircraft. Our preliminary results show that the computational times are excessive (more than one day for one iteration of an instance with 5613 scheduled flights and 145 pilots and copilots).

To compare the integrated and sequential approaches, we solved a personalized assignment problem for each of the pilots and copilots. The integrated approach gives better results in terms of the satisfaction of the flight and vacation preferences. For the vacations, see Figure 3; the percentage of preferences satisfied is the average over all three instances. The improvement is 5.25% for pilots and 4.14% for copilots when the flight bonus is -50 and the vacation penalty is 5000. We do not present an equivalent figure for the preferred flights because the improvement is small (less than 3% on average).



(a) Pilots (b) Copilots

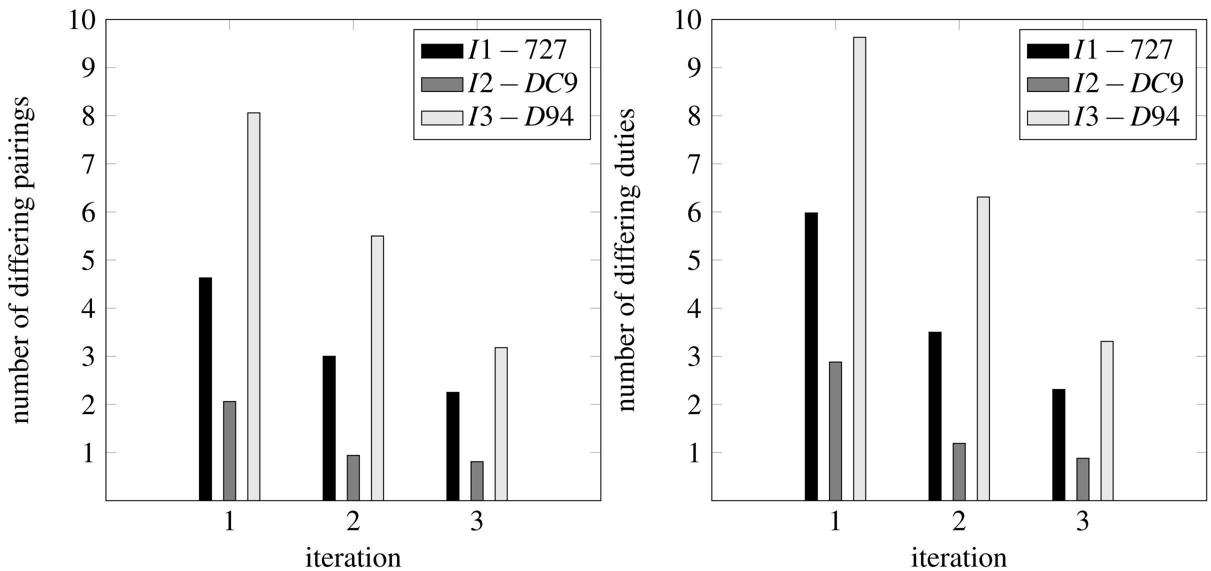
Figure 3: Sequential versus integrated

To evaluate the efficiency of performing several iterations in the heuristic algorithm, we compare the average number of differing pairings and duties over all the tests; see Figure 4. The results show that allowing several iterations leads to more robust schedules. Over the three iterations the number of pairings decreases by 2.38, 1.25, and 4.88 for instances $I1-727$, $I2-DC9$, and $I3-D94$, and the number of duties decreases by 3.67, 2.00, and 6.32.

For larger instances with 5000–7000 flights and 145–305 crew members, the size of the master problem increases. The numbers of subproblems, CG iterations, and branching nodes also increase. With appropriate strategies such as limits on the number of subproblems during the process, it will be possible to apply the heuristic algorithm to larger problems.

6 Conclusion

We have proposed a set partitioning formulation and a heuristic algorithm for the integrated personalized cockpit crew scheduling problem, which has not yet been investigated in the literature. Taking the crew preferences into account is common in European airlines and increasingly adopted in North America. We considered preferred flights and vacations. Our results show that the integrated approach satisfies more preferences than the sequential approach does. Furthermore, because of the inevitable perturbations, crew schedules need to be robust, and the integrated approach helps to provide more robust schedules by increasing the number of common pairings for pilots and copilots.



(a) Differing pairings

(b) Differing duties

Figure 4: Differing pairings and duties

References

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., and Vance, P.H. (1996). Branch-and-price: Column generation for solving huge integer programs. Operations Research, 46:316–329.

Cacchiani, V. and Salazar-González, J.J. (2013). A heuristic approach for an integrated fleet-assignment, aircraft-routing and crew-pairing problem. Electronic Notes in Discrete Mathematics, 41:391–398.

Cohn, A.M. and Barnhart, C. (2003). Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–396.

Cordeau, J.-F., Stojković, G., Soumis, F., and Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science, 35(4):375–388.

Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F. (1995). Time constrained routing and scheduling. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, Network Routing, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. Elsevier.

Desrosiers, J. and Lübbecke, M.E. (2005). A Primer in Column Generation. Springer.

Elhallaoui, I., Metrane, A., Soumis, F., and Desaulniers, G. (2010). Multi-phase dynamic constraint aggregation for set partitioning type problems. Mathematical Programming, 123(2):345–370.

Elhallaoui, I., Villeneuve, D., Soumis, F., and Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation. Operations Research, 53(4):632–645.

Gao, C., Johnson, E., and Smith, B. (2009). Integrated airline fleet and crew robust planning. Transportation Science, 43(1):2–16.

Guo, Y., Mellouli, T., Suhl, L., and Thiel, M.P. (2006). A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. European Journal of Operational Research, 171(3):1169–1181.

Irnich, S. and Desaulniers, G. (2005). Shortest Path Problems with Resource Constraints. Springer US.

Kasirzadeh, A. Saddoune, M., and Soumis, F. (2014). Airline crew scheduling: Models, algorithms, and data sets.

Klabjan, D., Johnson, E.L., Nemhauser, G.L., Gelman, E., and Ramaswamy, S. (2002). Airline crew scheduling with time windows and plane-count constraints. Transportation Science, 36(3):337–348.

Lohatepanont, M. and Barnhart, C. (2004). Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. Transportation Science, 38(1):19–32.

Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research, 32(6):1451–1476.

Papadakos, N. (2009). Integrated airline scheduling. Computers & Operations Research, 36(1):176–195.

Saddoune, M., Desaulniers, G., Elhallaoui, I., and Soumis, F. (2011). Integrated airline crew scheduling: A bi-dynamic constraint aggregation method using neighborhoods. European Journal of Operational Research, 212(3):445–454.

Saddoune, M., Desaulniers, G., Elhallaoui, I., and Soumis, F. (2012). Integrated airline crew pairing and crew assignment by dynamic constraint aggregation. Transportation Science, 46(1):39–55.

Saddoune, M., Desaulniers, G., and Soumis, F. (2013). Aircrew pairings with possible repetitions of the same flight number. Computers & Operations Research, 40(3):805–814.

Sandhu, R. and Klabjan, D. (2007). Integrated airline fleeting and crew-pairing decisions. Operations Research, 55(3):439–456.

Sherali, H.D., Bae, K.-H., and Haouari, M. (2013). A Benders decomposition approach for an integrated airline schedule design and fleet assignment problem with flight retiming, schedule balance, and demand recapture. Annals of Operations Research, 210(1):213–244.

Souai, N. and Teghem, J. (2009). Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. European Journal of Operational Research, 199(3):674–683.

Villeneuve, D. (1999). Logiciel de génération de colonnes. PhD thesis, Ecole Polytechnique de Montréal.

Weide, O., Ryan, D., and Ehrgott, M. (2010) An iterative approach to robust and integrated aircraft routing and crew scheduling. Computers & Operations Research, 37(5):833–844.

Zeghal, F. and Minoux, M. (2006). Modeling and solving a crew assignment problem in air transportation. European Journal of Operational Research, 175(1):187–209.