Les Cahiers du GERAD ISSN: 0711–2440

Combining alternating Lagrangian decomposition, column generation, and dynamic constraint aggregation for integrated crew pairing and personalized assignment problems simultaneous for pilots and copilots

V. Zeighami,
F. Soumis

G–2018–37

May 2018
Revised: November 2018

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, 2018
– Bibliothèque et Archives Canada, 2018

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, 2018
– Library and Archives Canada, 2018

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

Combining alternating Lagrangian decomposition, column Generation, and dynamic constraint aggregation for integrated crew pairing and personalized assignment problems for pilots and copilots Simultaneously

Vahid Zeighami

François Soumis

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

vahid.zeighami@gerad.ca
francois.soumis@gerad.ca

May 2018
Revised: November 2018
Les Cahiers du GERAD
G–2018–37

Abstract: The airline crew scheduling problem involves determining schedules for airline crew members such that all the scheduled flights over a planning horizon (usually a month) are covered and the constraints are satisfied. Because of its complexity, this problem is usually solved sequentially in two main steps: the crew pairing followed by the crew assignment. However, finding a globally optimal solution via the sequential approach may be impossible because the decision domain of the crew assignment problem is reduced by decisions made in the pairing problem. This study considers the crew scheduling problem in a personalized context where each pilot and copilot requests a set of preferred flights and vacations each month. We propose a model that completely integrates the crew pairing and personalized assignment problems to generate personalized monthly schedules for a given set of pilots and copilots simultaneously in a single optimization step. The model keeps the pairings in the two problems as similar as possible so that the propagation of perturbations arising during the operation is reduced. We develop an integrated algorithm that combines alternating Lagrangian decomposition, column generation, and dynamic constraint aggregation. We investigate conditions for the convergence of the algorithm, and we conduct computational experiments on a set of real instances from a major US carrier. Our integrated approach produces significant cost savings and better satisfaction of crew preferences compared with the traditional sequential approach.

Keywords: integrated crew scheduling problem, column generation, dynamic constraint aggregation, alternating Lagrangian decomposition

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.

1 Introduction

The airline crew scheduling problem is complex. For the airline industry, the crew costs are the largest after the fuel costs (Barnhart et al. 2003). Hence, even slight improvements in the quality of the schedule can have a significant financial benefit. Given a set of flights to be operated by the same aircraft fleet, defined by departure/arrival stations and fixed departure/arrival dates and times, the crew scheduling problem constructs individual schedules for a set of available crew members. Because of its complexity, this problem is usually solved in two steps: crew pairing followed by crew assignment.

A pairing is a sequence of flights, connections, and rests starting and ending at the same crew base. Given the scheduled flights, the crew pairing problem involves determining a set of feasible pairings such that the cost of the pairings is minimized and each flight is covered exactly once.

In the crew assignment problem, the goal is to build monthly schedules from these pairings for each individual crew member such that every pairing is covered exactly once while respecting all the safety and collective agreement rules. There are three different approaches for this problem. The bidline approach, used in some North American airlines, produces anonymous monthly schedules, which the crew members then select in seniority order. The personalized approach with strict seniority, which is becoming more popular in North America, sequentially maximizes the satisfaction of the employees in decreasing order of seniority. The personalized approach with a global objective, which is often employed by European airlines and has recently started to be used by American airlines, produces the monthly schedules simultaneously to maximize the sum of the personal satisfactions without any advantages for seniority.

The sequential approach to crew scheduling considerably reduces the complexity of the process but may lead to significantly suboptimal solutions since the schedule constraints and objectives are not taken into account during the construction of the pairings. In fact, the pairing construction stage cannot determine the best pairings for the crew assignment stage. Hence, it is difficult to maximize the satisfaction of the preferences at the assignment stage, and we may not find the globally optimal solution of the crew scheduling problem.

The integration of two or more other airline planning problems involving crew scheduling has been studied; see the recent survey by Kasirzadeh et al. (2017). Cordeau et al. (2001) and Mercier et al. (2005) proposed an algorithm based on column generation (CG) and Benders decomposition (BD) for integrated aircraft routing and crew pairing. Integrated flight scheduling, aircraft routing, and crew pairing was considered by Klabjan et al. (2002). Cohn and Barnhart. (2003) developed an extended crew pairing model that integrates crew scheduling and maintenance routing decisions. Sandhu and Klabjan (2007) introduced a model that completely integrates the fleeting and crew pairing stages. They proposed two approaches, the first based on a combination of Lagrangian relaxation and CG and the second a BD approach. Integrated aircraft routing, crew scheduling, and flight retiming was studied by Mercier and Soumis (2007). Papadakos (2009) introduced a set covering model for integrated fleet assignment, maintenance routing, and crew pairing and a method based on BD combined with CG. Gaoet et al. (2009) studied the integrated fleet and crew robust planning problem, providing fleet assignment solutions. Shao et al. (2015) considered integrated fleet assignment, aircraft routing, and crew pairing. Cacchiani and Salazar-González (2016) proposed two mixed integer linear programming models for integrated fleet assignment, aircraft routing, and crew pairing.

In recent decades, various models and methods have been introduced for the crew scheduling problem. Sequential approaches may lead to poor solutions, but only a few researchers have investigated integrated models. Zeghal and Minoux (2006) proposed an integer linear programming model using clique constraints for integrated pairing and bidline assignment. They consider small problems with 59 to 210 flights and a few flights per pairing. Guo et al. (2006) described a partially integrated crew scheduling approach based on pairing-chain generation. 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. Saddoune et al. (2012) developed a model and algorithm based on CG and dynamic constraint aggregation (DCA) for integrated crew pairing and assignment, where the objective is to minimize the total cost and the number of pilots. Kasirzadeh (2015) presented a heuristic algorithm based on CG and DCA for integrated crew pairing and personalized assignment.

Zeighami and Soumis (2017) proposed an integrated model for the crew pairing and personalized assignment problems and a method based on BD and CG. They considered a set of vacation requests (VRs) for each pilot and copilot each month. The pairings are generated by the Benders master problem, which is composed of a CG master problem (MMP) and a set of CG subproblems (MSP). The monthly schedules for pilots and copilots are generated by the Benders subproblems, which consist of a CG master problem (SMP) and a set of CG subproblems (SSP). The Benders cuts contain pairing variables. The challenge is that the dual values of the unknown pairings from SMP do not exist, and for known pairings there are no arcs corresponding to pairings in the MSP to transfer the information on the Benders cuts from the restricted MMP to the MSP. To overcome this difficulty, they proposed weak Benders cuts that do not take into account the dual variables corresponding to the pairings. They proved that with these weak Benders cuts and the suggested dual solution of the Benders subproblems, the Benders approach reaches optimality. In a more general airline crew scheduling problem, each pilot and copilot can choose a set of preferred flights (PFs) from the monthly schedule. Since the information on PFs is included in the pairings and in the proposed weak Benders cuts of Zeighami and Soumis (2017), the dual value of each pairing is zero, and it is not possible to transfer information on the PFs from the Benders subproblems to the master problem. Thus, Zeighami and Soumis (2017) cannot deal with PFs. The aim of this study is to fill this gap.

We consider the integrated pilot and copilot crew pairing and personalized crew assignment problems in a personalized context with a global objective. Each flight must be covered by one pilot and one copilot, and they request a set of VRs and PFs each month. The pilots and copilots have different schedules to maximize the satisfaction of their preferences. However, to reduce the propagation of perturbations arising during the operation, their pairings must be as similar as possible. We therefore optimize the schedules for the pilots and copilots simultaneously, taking their preferences into account; see Figure 1. For simplicity, we use the term *integrated pilot and copilot problem* to refer to the integrated pilot and copilot crew pairing and personalized crew assignment problem.

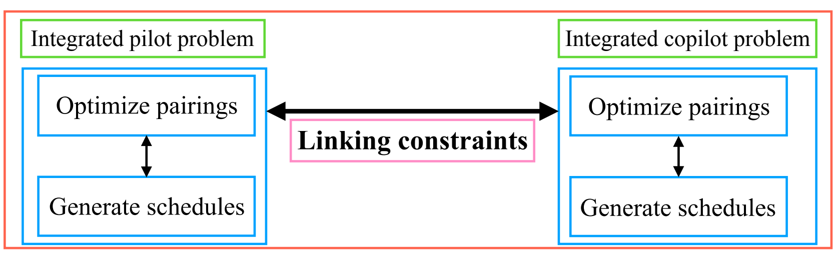

Figure 1: Integrated pilot and copilot problems optimized simultaneously within a comprehensively integrated framework

The contributions of this study are as follows: we introduce a novel integrated crew pairing and personalized assignment model to generate personalized monthly schedules for a given set of pilots and copilots simultaneously. To decrease the propagation of perturbations, we keep the pairings in the two problems as similar as possible. To solve this integrated model, we develop an algorithm based on alternating Lagrangian decomposition (ALD), CG, and DCA. The approach uses Lagrangian decomposition in an alternating fashion by proposing a new way to update the Lagrangian multipliers. The solution process iterates between pilots and copilots by estimating the effects of decisions made in one problem on the other. Also, we study the conditions for convergence. The computational results reveal that our approach outperforms the sequential approach in terms of cost savings and the satisfaction of crew preferences. Moreover, the computational time required is similar.

The remainder of this paper is organized as follows. Section 2 gives the problem statement, and Section 3 gives the mathematical formulation. The algorithm is outlined in Section 4, and Section 5 gives our experimental results and observations. Finally, Section 6 provides concluding remarks.

2 Problem statement

In this section, we give the statement of first the crew pairing problem and then the personalized crew assignment problem. To define a pairing and a schedule we use the subset of feasibility rules given by Zeighami and Soumis (2017).

Each flight is defined by a flight number, departure/arrival airports, and fixed departure/arrival dates and times. Consider a set of flights and crew bases that must be operated by the same aircraft fleet over a planning horizon (usually one month). The crew pairing problem finds a set of minimum-cost pairings such that each flight is covered by exactly one pairing. A pairing is a sequence of one or more duties and overnight stops. A duty is a working day for a crew member and consists of a sequence of consecutive flights or deadheads separated by connection periods. A deadhead is a flight where the crew travels as passengers to be relocated. A pairing is feasible if it satisfies the following rules. There is a maximum of 5 landings per duty. The briefing and debriefing times at the beginning and end of each duty are 60 and 30 minutes, respectively. The maximum pairing duration is 4 days, the maximum number of duties per pairing is 4, the minimum connection time between two consecutive flights in a duty is 30 minutes, and the minimum connection time between two consecutive duties is 9.5 hours. A crew member must work 4 to 8 hours in a duty, a minimum of 4 hours is paid even if they are not worked, and the length of a duty cannot exceed 12 hours. The cost of a pairing has a complicated structure with three components: the cost of waiting times, the deadheading cost, and the total cost of the duties in the pairing. We use the pairing cost of Saddoune et al. (2012).

Pilots and copilots are trained for a specific aircraft fleet, and they are associated with a crew base. All pairings assigned to a pilot or copilot must start and end at his/her base. A base is a large airport where pilots and copilots are stationed (with an equal number of each rank); we assume that the number at each base is fixed. In our test, each pilot and copilot requests a set of PFs and VRs per month. A schedule is a sequence of pairings separated by rest periods. We build monthly schedules that cover all the pairings previously built, and each flight must be covered by one pilot and one copilot. A schedule is feasible if it satisfies the following safety and collective agreement rules. There is a maximum of 85 flying hours per month and a maximum of 6 consecutive working days. In each schedule, the pairings are separated by two different rest times: the day-off rest and the rest between two consecutive pairings. The minimum rest time between any two consecutive pairings is 8 hours. A penalty cost is associated with each unsatisfied VR, and a negative cost is associated with each satisfied PF. The objective function maximizes the number of satisfied VRs and PFs.

3 Mathematical formulation

In this section, we present our integrated crew pairing and personalized crew assignment model for a given set of pilots and copilots within a completely integrated framework. The model constructs the pairings and monthly schedules for each pilot and copilot in a single step directly from flights instead of pairings. We keep the pairings in the two problems as similar as possible, while satisfying the feasibility rules impacting the pairings and monthly schedules. We use the following notation. Let F be the set of scheduled flights to be covered. Let the set of pilots be L and the set of copilots O. We denote by V_l the set of VRs for pilot $l \in L$ and by V_o the set of VRs for copilot $o \in O$. Let S_l be the set of feasible schedules for pilot $l \in L$ and S_o the set of feasible schedules for copilot $o \in O$. The binary variable x_s is equal to 1 if schedule $s \in S_l$ is allocated to pilot $l \in L$ and 0 otherwise, and y_s is equal to 1 if schedule $s \in S_o$ is allocated to copilot $o \in O$ and 0 otherwise. The variable r_v is 1 if VR $v \in V_l$ is satisfied for pilot $l \in L$ and 0 otherwise, and z_v is 1 if VR $v \in V_o$ is satisfied for copilot $o \in O$ and 0 otherwise. Let c_v be the penalty cost for unsatisfied VR $v \in V_l$ (or $v \in V_o$). The binary constant e_f^s is equal to 1 if flight $f \in F$ is covered by schedule $s \in S_l$ (or $s \in S_o$). The binary constant b_v^s is equal to 1 if VR $v \in V_l$ (or $v \in V_o$) is satisfied by schedule $s \in S_l$ (or $s \in S_o$). Let n_s be the number of PFs in schedule $s \in S_l$ (or $s \in S_o$), and let B be the bonus cost (a negative cost) for covering each PF. We denote by c_p the cost of pairing p. Let c_s be the cost of schedule $s \in S_l$ (or $s \in S_o$), where $c_s = \sum_{p \in P_s} c_p + Bn_s$, and P_s is the set of all pairings covered by schedule s. Let A be all the feasible connection arcs between two successive flights. For every connection arc $(i, j) \in A$ and every schedule s, we set the binary constant a_{ij}^s to 1 if (i,j) between flights i and j is covered by s. The variable u_{ij} is 1 if $(i,j) \in A$ is covered by the pilot problem or copilot problem but not both, and 0 otherwise. Let R_{ij} be the penalty cost for $(i,j) \in A$ if u_{ij} is equal to 1. The value of the penalty cost R_{ij} is inversely proportional to the connection time between flights i and j. This forces the model to include these flights in the same pairing in both the pilot and copilot problems where possible.

Using this notation, the integrated model is as follows:

$$\min \sum_{l\in L}\sum_{s\in S_l} c_s x_s + \sum_{l\in L}\sum_{v\in V_l} c_v r_v + \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v + \sum_{(i,j)\in A} R_{ij} u_{ij} \tag{1}$$

Integrated pilot constraints:

$$\sum_{l\in L}\sum_{s\in S_l} e_f^s x_s = 1 \qquad \forall f \in F \tag{2}$$

$$\sum_{s\in S_l} b_v^s x_s + r_v = 1 \qquad \forall l \in L, \forall v \in V_l \tag{3}$$

$$\sum_{s\in S_l} x_s \leq 1 \qquad \forall l \in L \tag{4}$$

Linking constraints:

$$\sum_{l\in L}\sum_{s\in S_l} a_{ij}^s x_s - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s \leq u_{ij} \qquad \forall (i,j) \in A \tag{5}$$

$$\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - \sum_{l\in L}\sum_{s\in S_l} a_{ij}^s x_s \leq u_{ij} \qquad \forall (i,j) \in A \tag{6}$$

Integrated copilot constraints:

$$\sum_{o\in O}\sum_{s\in S_o} e_f^s y_s = 1 \qquad \forall f \in F \tag{7}$$

$$\sum_{s\in S_o} b_v^s y_s + z_v = 1 \qquad \forall o \in O, \forall v \in V_o \tag{8}$$

$$\sum_{s\in S_o} y_s \leq 1 \qquad \forall o \in O \tag{9}$$

$$x_s \in \{0,1\} \qquad \forall l \in L, \forall s \in S_l \tag{10}$$

$$r_v \in \{0,1\} \qquad \forall l \in L, \forall v \in V_l \tag{11}$$

$$u_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{12}$$

$$y_s \in \{0,1\} \qquad \forall o \in O, \forall s \in S_o \tag{13}$$

$$z_v \in \{0,1\} \qquad \forall o \in O, \forall v \in V_o \tag{14}$$

The objective function (1) finds a trade-off between maximizing the number of satisfied VRs and PFs and minimizing the total cost of the pairings and the dissimilarity of the pilot and copilot pairings. Set-partitioning constraints (2) and (7) ensure that each flight is covered by exactly one pilot and one copilot. Constraints (3) and (8) impose the VRs for each pilot and copilot respectively. Constraints (4) and (9) guarantee that at most one schedule is chosen for each pilot and copilot respectively. Linking constraints (5) force variable u_{ij} to be 1 if $(i,j) \in A$ is covered only in the pilot problem. Linking constraint (6) forces variable u_{ij} to be 1 if $(i,j) \in A$ is covered only in the copilot problem. The integrality conditions are defined by constraints (10)–(14).

Remark 1 *The presence of differently covered connection arcs in the two problems leads to different pairings.*

4 Solution methodology

In this section, we describe our algorithm. The integrated model (1)–(14) contains a large number of linking constraints; see Table 3. To handle these constraints, we propose a solution approach based on ALD. The algorithm iterates between two Lagrangian subproblems, the first for the pilots and the second for the copilots. In both subproblems, the number of variables grows exponentially as the number of flights increases, so we use a CG method. However, because of degeneracy, CG becomes inefficient when the number of set partitioning constraints is large and the columns are dense (more than 8–12 nonzero elements per column; see Elhallaoui et al. (2005)). In our problem, there are between 30 and 45 nonzeros per column. To overcome this difficulty and accelerate the solution process, we combine CG with DCA. In the following subsections, we explain our approach and then describe how to apply Lagrangian decomposition to model (1)–(14). Finally, we describe the integrated approach.

4.1 Column generation

CG (Barnhart et al. 1998) is an iterative method for large problems with many variables. CG decomposes the main problem into a restricted master problem (RMP) and several pricing problems. The RMP is a restriction of the master problem and contains a subset of the columns. New variables for the master problem are then generated at each iteration by solving pricing problems, with the goal of improving the RMP objective function. At each CG iteration, we solve the RMP using an LP solver to produce primal and dual solutions. Based on the dual solution, we solve the pricing problems to find negative-reduced-cost variables. We then add these variables to the current RMP. We iterate until no negative-reduced-cost variables are identified: the current RMP primal solution is then optimal for the master problem. In our case, each pricing problem corresponds to a resource-constrained shortest path problem that is solved using a label-setting algorithm (Desrochers and Soumis. 1988).

4.1.1 Pricing problems

There is one pricing problem for each pilot and copilot, defined on a directed acyclic time-space network. The network has six node types: *source, sink, departure, arrival, midnight*, and *waiting*. The source and sink nodes represent the start and end of the schedules. Each flight is defined by departure and arrival nodes, and there is a waiting node for each departing node. The midnight nodes are used to define the start and end of each day. The network has twelve arc types: *start of schedule, end of schedule, flight, deadhead, VR, rest, wait time, start of duty, start of pairing, day off, post-pairing, post-pairing rest*. The start-of-schedule arc connects the source node to the first midnight node of the horizon. Each flight arc and deadhead arc is defined by a departure/arrival node and fixed departure/arrival dates and times. A deadhead arc has a fixed cost for each occurrence of deadheading in a pairing and a variable cost that depends on the length of the deadhead. A short-rest arc links the arrival node of a flight to the departure nodes of all the flights at the same base if the time interval is between the minimum and the ideal maximum rest time. If the waiting time exceeds the ideal maximum rest time, a long-rest arc connects the arrival node of the flight to the earliest waiting node. Waiting arcs either link two consecutive waiting nodes to extend the long-rest duration or connect two consecutive flights in a duty. An empty arc links each waiting node to its corresponding departure node. A rest arc links an end-of-pairing node at the base to the start node of a pairing whose departure time is greater than or equal to the arrival time plus the minimum rest time (8 hours in our tests). A start-of-day-off arc links an end-of-pairing node to the first midnight node at the base to start a day off. A day-off arc connects a pair of consecutive midnight nodes at the base. A VR arc links two midnight nodes covering between one and ten days. To ensure schedule feasibility, eight resources are used: maximum pairing duration, maximum number of duties in a pairing, maximum number of landings per duty, maximum working time per duty, maximum total duty duration, minimum number of days off, maximum number of consecutive working days, and maximum credited flying time.

4.1.2 Labeling algorithm

The pricing problems are solved using a labeling algorithm. Every feasible schedule corresponds to a path from the source node to the sink node in the above network. The feasibility constraints on the schedules are enforced via resource constraints in the network. For each feasibility constraint, there is a resource window at each node, and the value of every constrained resource must be within this interval. Resources are consumed on arcs of the network. The cost of a feasible path corresponding to a schedule from the source node to the sink node is the sum of the arc costs on the path. The arc costs are updated during the solution process based on the dual variables for the master problem constraints. A label is associated with each partial path starting from the source node. Each label has a set of resource components and a cost component. The resource components determine the value of the resources at the last node of the partial path, and the cost component calculates the reduced cost of this path. To reduce the time for the path generation, we eliminate labels using a dominance rule: label L_1 is dominated by label L_2 if the resource components and cost component of L_1 are less than or equal to the corresponding components of L_2. We use a heuristic and an exact version of the labeling algorithm. The heuristic version considers a subset of the resource components with the reduced-cost component in the dominance rule. If no negative-reduced-cost paths are found, the exact version uses all the resource components.

4.2 Dynamic Constraint Aggregation

DCA (Elhallaoui et al. 2005) is a new version of CG that reduces the number of set-partitioning constraints in the RMP by aggregating some of them. The approach aggregates clusters of the RMP set-partitioning constraints and retains one representative constraint for each cluster. DCA relies on an aggregated restricted master problem (ARMP) that considers smaller subsets of variables and constraints compared to traditional RMP. This new version speeds up the solution process. At the beginning of DCA, a partition is defined with respect to an initial set of clusters that is provided by a planned solution. During the CG solution process, this aggregation can change dynamically. A column is said to be compatible with the current partition if, for each cluster of the partition, it covers either all of the cluster's tasks or none. Otherwise, this column is incompatible. A newly generated column can be added to ARMP if it is compatible with the current partition. An incompatible column can be added to the ARMP only if the partition is modified. The ARMP is solved using the simplex algorithm to produce a primal solution and a dual solution for the aggregated constraints. To compute the dual values for each set partitioning constraint of the original problem, we use a dual variable disaggregation procedure. A modified version of DCA was developed by Elhallaoui et al. (2010) with multiple phases (MPDCA). MPDCA defines an incompatibility number (r) with respect to the current partition for each column. It estimates the minimum number of additional clusters needed to make an incompatible column compatible. In phase k, only the incompatible variables with $r \leq k$ are priced out. 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. All variables with 0-incompatibilities for $0 \leq k$ are priced out by solving the pricing problems. When there is no negative-reduced-cost column, the algorithm continues to phase $(k+1)$ or stops if k is the last phase. 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 some or all of the compatible columns to the ARMP, and the MPDCA moves to its next iteration. Otherwise, we perform a test to determine whether or not the current partition should be changed. We update the partition if the reduced cost of the least-reduced-cost compatible column is greater than or equal to the reduced cost of the least-reduced-cost incompatible column times a predetermined multiplier.

4.3 Alternating Lagrangian decomposition

To reduce the complexity of the solution process, model (1)–(14) can be decomposed into two subproblems via Lagrangian decomposition (Guignard and Kim. 1987) and solved by combining CG and DCA. The two subproblems are called ALD-pilot and ALD-copilot. The classical Lagrangian decomposition using subgradient or bundle methods usually needs many iterations to reach an optimal solution. We use Lagrangian decomposition in an alternating fashion by proposing a new way to update the Lagrangian multipliers. The algorithm iterates between ALD-pilot and ALD-copilot by estimating the effect of decisions made in one problem on the other. Without loss of generality, we first solve the integrated pilot model (2)–(4) considering the first two terms of function (1) as the objective function; let $\bar{x}_s$ ($s \in S_l, l \in L$) be the optimal solution. Let $A^1_L = \{(i,j) \in A | \sum_{l\in L} \sum_{s\in S_l} a^s_{ij}\bar{x}_s = 1\}$ and $A^0_L = \{(i,j) \in A | \sum_{l\in L} \sum_{s\in S_l} a^s_{ij}\bar{x}_s = 0\}$ be the set of connection arcs that are covered and uncovered, respectively. The resulting linking constraints (5)–(6) are

$$1-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s \le u_{ij} \qquad \forall (i,j)\in A^1_L\subseteq A \quad (\text{multipliers } (\lambda_{ij})_{pilot}\ge 0) \tag{15}$$

$$\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - 1 \le u_{ij} \qquad \forall (i,j)\in A^1_L\subseteq A \tag{16}$$

$$0-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s \le u_{ij} \qquad \forall (i,j)\in A^0_L\subseteq A \tag{17}$$

$$\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - 0 \le u_{ij} \qquad \forall (i,j)\in A^0_L\subseteq A \quad (\text{multipliers } (\pi_{ij})_{pilot}\ge 0) \tag{18}$$

Lemma 1 *Constraints (16) and (17) are redundant for the integrated copilot problem.*

Proof. Proof. Clearly $0 \le \sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s \le 1$, so $-1 \le \sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - 1 \le 0$ and $-1 \le -\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s \le 0$. Since $u_{ij}\ge 0$, constraints (16) and (17) are redundant. □

Let $\boldsymbol{(\lambda)_{Pilot}} = \{(\lambda_{ij})_{pilot} \ge 0 | (i,j)\in A^1_L\}$ and $\boldsymbol{(\pi)_{Pilot}} = \{(\pi_{ij})_{pilot} \ge 0 | (i,j)\in A^0_L\}$ be Lagrangian multipliers associated with constraints (15) and (18), respectively, and calculated from the pilot problem as explained in Section 4.3.1. The Lagrangian decomposition is obtained by dualizing these linking constraints. The resulting copilot subproblem is

$$\begin{aligned}&\text{ALD-copilot } (\lambda,\pi)_{Pilot} = \min \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v + \sum_{(i,j)\in A} R_{ij}u_{ij} + \\ &\sum_{(i,j)\in A^1_L} (\lambda_{ij})_{pilot}(1-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - u_{ij}) + \sum_{(i,j)\in A^0_L} (\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - 0 - u_{ij})\end{aligned} \tag{19}$$

subject to (7)–(9) and (12)–(14).

After we solve ALD-copilot and fix variable y_s in model (1)–(14), the resulting linking constraints (5)–(6) are

$$\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}x_s - 1 \le u_{ij} \qquad \forall (i,j)\in A^1_O\subseteq A \tag{20}$$

$$1-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}x_s \le u_{ij} \qquad \forall (i,j)\in A^1_O\subseteq A \quad (\text{multipliers } (\lambda_{ij})_{copilot}\ge 0) \tag{21}$$

$$\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}x_s - 0 \le u_{ij} \qquad \forall (i,j)\in A^0_O\subseteq A \quad (\text{multipliers } (\pi_{ij})_{copilot}\ge 0) \tag{22}$$

$$0-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}x_s \le u_{ij} \qquad \forall (i,j)\in A^0_O\subseteq A \tag{23}$$

Here $A^1_O = \{(i,j)\in A | \sum_{o\in O}\sum_{s\in S_o} a^s_{ij}\bar{y}_s = 1\}$ and $A^0_O = \{(i,j)\in A | \sum_{o\in O}\sum_{s\in S_o} a^s_{ij}\bar{y}_s = 0\}$ are the sets of connection arcs that are covered and uncovered by ALD-copilot respectively, and $\bar{y}_s$ $(s\in S_o; o\in O)$ is the optimal solution.

Lemma 2 *Constraints (20) and (23) are redundant for the integrated pilot problem.*

The proof is similar to that for Lemma 1.

Now let $\boldsymbol{(\lambda)_{Copilot}} = \{(\lambda_{ij})_{copilot} \ge 0 | (i,j)\in A^1_O\}$ and $\boldsymbol{(\pi)_{Copilot}} = \{(\pi_{ij})_{copilot} \ge 0 | (i,j)\in A^0_O\}$ be Lagrangian multipliers associated with constraints (21) and (22), respectively, and calculated from ALD-copilot. The resulting pilot subproblem is

$$\text{ALD-pilot } (\lambda,\pi)_{Copilot} = \min \sum_{l\in L}\sum_{s\in S_l} c_s x_s + \sum_{l\in L}\sum_{v\in V_l} c_v r_v + \sum_{(i,j)\in A} R_{ij}u_{ij} +$$
$$\sum_{(i,j)\in A_O^1} (\lambda_{ij})_{copilot}(1 - \sum_{l\in L}\sum_{s\in S_l} a_{ij}^s x_s - u_{ij}) + \sum_{(i,j)\in A_O^0} (\pi_{ij})_{copilot}(\sum_{l\in L}\sum_{s\in S_l} a_{ij}^s x_s - 0 - u_{ij}) \qquad (24)$$

subject to (2)–(4) and (10)–(12).

4.3.1 Lagrangian multipliers

To compute the Lagrangian multipliers, we introduce the concept of an arc reduced cost to benefit from information connecting the two problems. In a traditional CG, the arc reduced costs are not directly available, and we require additional information from the pricing problem to compute them. Consider two consecutive flights i and j. We calculate the arc reduced cost on the arc (i,j) between these two flights using the following formula:

$$\bar{c}_{ij} = \Pi_j - \Pi_i. \qquad (25)$$

Here Π_i and Π_j are the minimum costs from the source node to the arrival node of flight i and the departure node of flight j, respectively. Π_i and Π_j are calculated using the labeling algorithm described in Section 4.1.2. Consider the example shown in Figure 2 (for clarity, some arcs and labels are truncated or omitted). Assume that we have three pilots: A, B, and C. Without loss of generality, we assume that in the optimal solution of ALD-pilot, flights i, j, and k are covered by pilot A; flights m, n, o, and p are covered by pilot B; and flights q, s, and r are covered by pilot C. Let Π_i^A and Π_j^A respectively be the minimum cost from the source node to the arrival node of flight i and the departure node of flight j, obtained from the pricing problem corresponding to pilot A. The arc reduced cost on (i,j) is then given by $\bar{\Pi}_j^A - \bar{\Pi}_i^A$; see Figure 3.

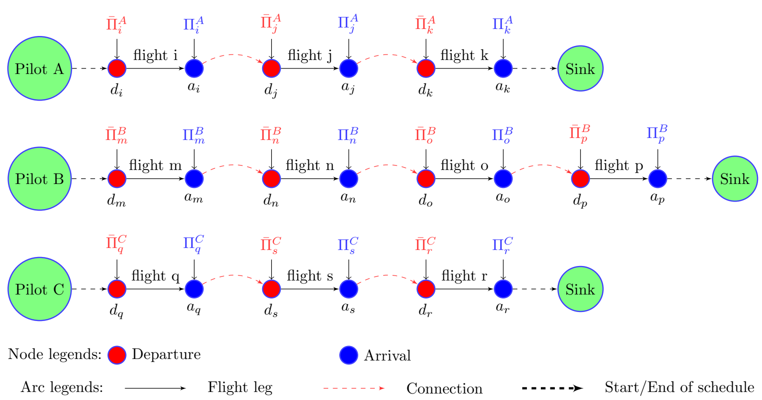

Figure 2: Finding minimum cost from source node to each departure and arrival node in pilot pricing problems

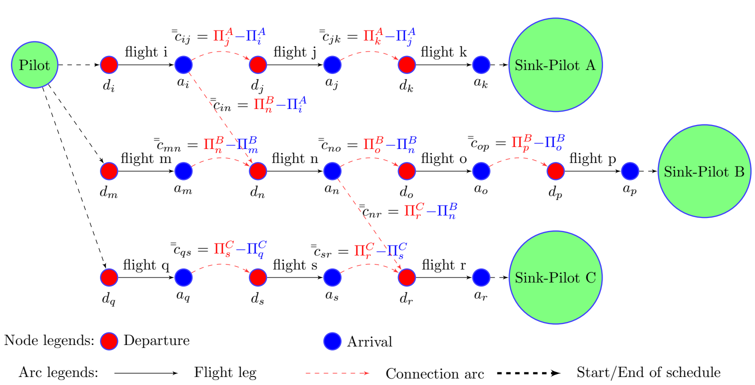

Figure 3: Calculate arc reduced cost on each connection arc in pilot pricing problems

Remark 2 *There is one pricing problem for each pilot (or copilot), so arc $(i,j) \in A$ may exist in more than one pricing problem. Therefore, there is more than one way to calculate the arc reduced cost on $(i,j) \in A$. We use the pilot (copilot) who covered the first flight and the pilot (copilot) who covered the second flight related to this arc in the optimal solution.*

Let $(\bar{c}_{ij})^k_{pilot}$ be the arc reduced cost of $(i,j) \in A$ obtained by ALD-pilot at iteration k. We pay a minimum penalty cost to ALD-copilot to have pairings similar to those obtained by ALD-pilot. For each $(i,j) \in A$, we calculate the Lagrangian multipliers as the minimum of the arc reduced cost $(\bar{c}_{ij})^k_{pilot}$ and the penalty cost R_{ij}. The Lagrangian multipliers for ALD-pilot (ALD-copilot) at iteration k are computed as follows:

$$(\lambda_{ij})^k_{pilot} = min\{max\{(\bar{c}_{ij})^k_{pilot}, 0\}, R_{ij}\} \qquad \forall (i,j) \in A^1_L \tag{26}$$

$$(\pi_{ij})^k_{pilot} = min\{max\{(\bar{c}_{ij})^k_{pilot}, 0\}, R_{ij}\} \qquad \forall (i,j) \in A^0_L \tag{27}$$

The Lagrangian multipliers can be considered dual information. Before we solve ALD-copilot, we must project the calculated multipliers on each connection arc in ALD-pilot onto the corresponding arcs in the copilot's pricing problems. We then solve ALD-copilot using CG combined with DCA. Since $(\lambda_{ij})_{pilot}$ is negated in the arc covering term of ALD-copilot (i.e., $(\lambda_{ij})_{pilot}(1 - \sum_{o\in O}\sum_{s\in S_o} a^s_{ij} y_s - u_{ij})$) we add the value $-(\lambda_{ij})_{pilot}$ on the arc (i,j) in all the ALD-copilot's pricing problems that include this arc. On the other hand, since $(\pi_{ij})_{pilot}$ is positive in the arc covering term of ALD-copilot (i.e., $(\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a^s_{ij} y_s - 0 - u_{ij})$) we add the value $+(\pi_{ij})_{pilot}$ on the arc (i,j) in all the ALD-copilot's pricing problems that include this arc; see Figure 4. In fact, we add a bonus cost $-(\lambda_{ij})_{pilot}$ to $(i,j) \in A$ in ALD-copilot's pricing problems if the arc is covered by ALD-pilot, and otherwise we add the penalty cost $+(\pi_{ij})_{pilot}$; see Figure 4. This encourages ALD-copilot to cover the connection arcs that are covered by ALD-pilot. The same strategy is used for ALD-pilot.

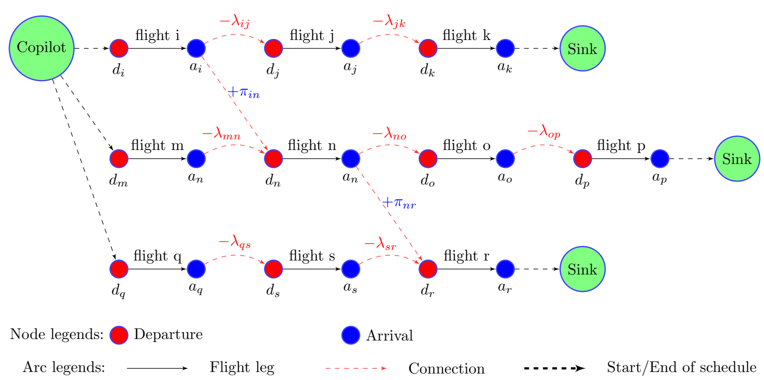

Figure 4: Projection of Lagrangian multipliers for pilot onto corresponding arcs in copilot pricing problems

Remark 3 *Let $(\bar{\bar{\bar{C}}}_{ij})^k_{copilot}$ be the Lagrangian arc reduced cost of ALD-copilot, calculated via*

$$(\bar{\bar{\bar{C}}}_{ij})^k_{copilot} = \begin{cases} (\bar{c}_{ij})^k_{copilot} - (\lambda_{ij})^k_{pilot}, & \textit{if } (i,j) \in A^1_L \\ (\bar{c}_{ij})^k_{copilot} + (\pi_{ij})^k_{pilot}, & \textit{if } (i,j) \in A^0_L. \end{cases}$$

Since $(\lambda_{ij})^k_{pilot}$ and $(\pi_{ij})^k_{pilot}$ are known, we can calculate $(\bar{c}_{ij})^k_{copilot}$ at each iteration to update the Lagrangian multipliers. The same is true for ALD-pilot.

We now have all the ingredients to derive an iterative algorithm for model (1)–(14).

4.4 Algorithm

In this section, we describe the integrated approach (INT). INT solves the Lagrangian subproblems by alternating between CG and DCA and transferring *primal* and *dual information* between the two subproblems.

Figure 5 illustrates the INT approach; the green line represents the Lagrangian terms in the objective function of ALD-copilot obtained via the Lagrangian multipliers from ALD-pilot.

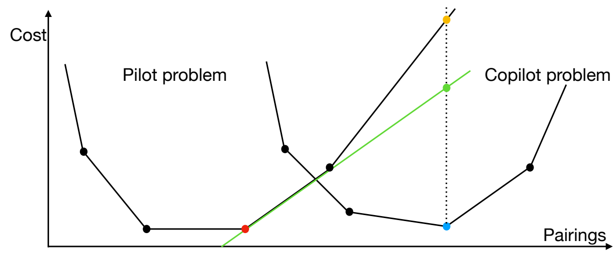

Figure 5: Schematic of INT approach

Primal information : To apply DCA to each subproblem we need to define a set of clusters as an initial partition. We define a cluster as a nonempty subset of consecutive flights. A partition is a set of clusters that covers all the flights; in our case, a cluster is a pairing. At each INT iteration, we use the optimal solution obtained by ALD-pilot as an initial partition for ALD-copilot, and vice versa.

Dual information : At each INT iteration, to estimate the effect of decisions made in ALD-pilot on ALD-copilot and define the Lagrangian terms in the objective function, ALD-copilot calculates the multipliers $(\lambda, \pi)_{pilot}$ and information about the coverage of each $(i, j) \in A$ (A_l^1, A_l^0) from ALD-pilot. ALD-pilot must do likewise.

Figure 6 gives a flowchart for INT. INT starts with the set of initial pairings constructed by Zeighami and Soumis (2017) as an initial partition. The initial values of the multipliers are 0. The algorithm starts by solving ALD-pilot using CG and DCA and constructs personalized monthly schedules for the pilots. It then solves ALD-copilot using CG and DCA and taking into account the primal and dual information from ALD-pilot, and it constructs personalized monthly schedules for the copilots. It alternates between the two subproblems until the relative difference between the lower and upper bounds is less than or equal to a predetermined threshold.

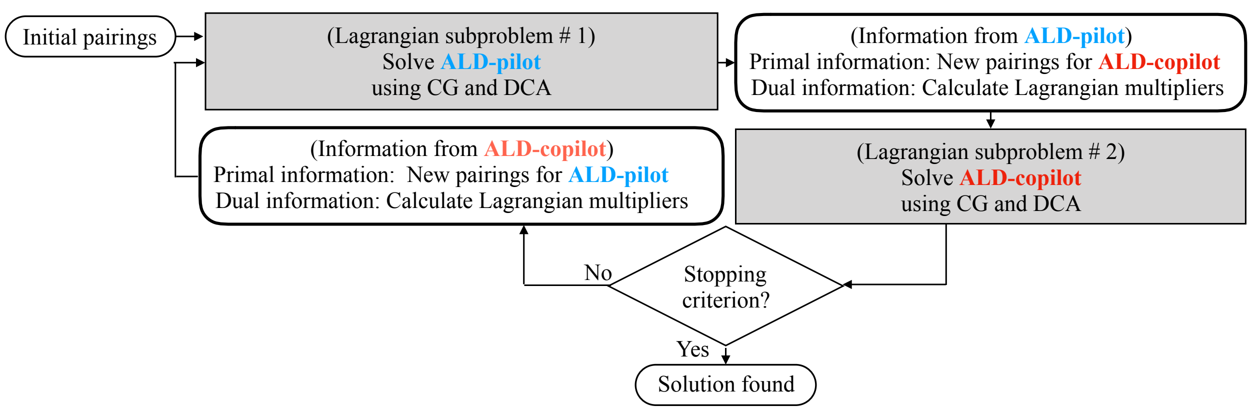

Figure 6: Flowchart of INT approach

4.4.1 Lower and upper bounds

Computing lower and upper bounds allows us to assess the quality of the solutions generated by the INT approach. The upper bounds can be computed at each iteration by using the solutions of ALD-pilot $(\bar{x}_s, \bar{r}_v)$ and ALD-copilot $(\bar{y}_s, \bar{z}_v)$ that are calculated during INT. Clearly, $(\bar{x}_s, \bar{r}_v)$ satisfies constraints (2)–(4) and $(\bar{y}_s, \bar{z}_v)$ satisfies constraints (7)–(9). By substituting $\bar{x}_s$ and $\bar{y}_s$ into linking constraints (5) and (6), we obtain a value of variable u_{ij} that is feasible for model (1)–(14). Using this feasible solution, i.e., $(\bar{x}_s, \bar{r}_v, \bar{y}_s, \bar{z}_v, \bar{u}_{ij})$, we compute an upper bound. To compute lower bounds we need the following definition.

Definition 1 *Let $C(y_s, z_v)$ be the cost function of ALD-copilot and $P(x_s^*, r_v^*)$ the optimal cost of ALD-pilot considering the first two terms of objective functions (19) and (24), respectively. Let $L(y_s)_{(\lambda,\pi)_{pilot}}$ be the cost function of the Lagrangian term obtained by solving ALD-copilot with respect to the Lagrangian multipliers obtained from ALD-pilot. Let $LB(y_s, z_v)_{(\lambda,\pi)_{pilot}} = P(x_s^*, r_v^*) + L(y_s)_{(\lambda,\pi)_{pilot}} + C(y_s, z_v)$.*

Proposition 1 *$LB(y_s, z_v)_{(\lambda,\pi)_{pilot}}$ is a lower bound on the optimal objective function value of model (1)–(14).*

Proof. Proof. $C(y_s, z_v)$ is the cost of the integrated copilot problem, and $P(x_s^*, r_v^*)+L(y_s)_{(\lambda,\pi)_{pilot}}$ (the green line in Figure 5) is a lower bound on the cost of the integrated pilot problem and the penalty cost of the dissimilarity of pairings between the two problems. Therefore, $LB(y_s, z_v)_{(\lambda,\pi)_{pilot}}$ is a lower bound on the optimal objective function value of model (1)–(14); see Figure 5. □

Definition 2 *Let $P(x_s, r_v)$ be the cost function of ALD-pilot and $C(y_s^*, z_v^*)$ the optimal cost of ALD-copilot considering the first two terms of objective functions (24) and (19), respectively. Let $L(x_s)_{(\lambda,\pi)_{copilot}}$ be the cost function of the Lagrangian term obtained by solving ALD-pilot with respect to the Lagrangian multipliers obtained from ALD-copilot. Let $LB(x_s, r_v)_{(\lambda,\pi)_{copilot}} = P(x_s, r_v) + L(x_s)_{(\lambda,\pi)_{copilot}} + C(y_s^*, z_v^*)$.*

Proposition 2 *$LB(x_s, r_v)_{(\lambda,\pi)_{copilot}}$ is a lower bound on the optimal objective function value of model (1)–(14).*

Proof. Proof. See the proof of Proposition 1. □

In the following two propositions, we show that INT converges to an optimal solution of model (1)–(14) if the similarity of the pairings between the two problems is 100%.

Proposition 3 *The following model is equivalent to the ALD-copilot model (19):*

$$\begin{aligned} &\min \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v + \sum_{(i,j)\in A_L^1} (\lambda_{ij})_{pilot}(1 - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s) + \sum_{(i,j)\in A_L^0} (\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - 0) \\ &\quad \textit{subject to } (7)\text{–}(9) \textit{ and } (12)\text{–}(14). \end{aligned} \tag{28}$$

Proof. Proof. Since $A = A_L^1 \cup A_L^0$, $\sum_{(i,j)\in A_L} R_{ij}u_{ij} = \sum_{(i,j)\in A_L^1} R_{ij}u_{ij} + \sum_{(i,j)\in A_L^0} R_{ij}u_{ij}$. Also,

$$\sum_{(i,j)\in A_L^1} (\lambda_{ij})_{pilot}(1 - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - u_{ij}) = \sum_{(i,j)\in A_L^1} (\lambda_{ij})_{pilot}(1 - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s) - \sum_{(i,j)\in A_L^1} (\lambda_{ij})_{pilot} u_{ij} \tag{29}$$

and

$$\sum_{(i,j)\in A_L^0} (\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - 0 - u_{ij}) = \sum_{(i,j)\in A_L^0} (\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s) - \sum_{(i,j)\in A_L^0} (\pi_{ij})_{pilot} u_{ij}. \tag{30}$$

By (29) and (30), the following model is equivalent to the ALD-copilot model (19):

$$\begin{aligned} \min &\sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v + \sum_{(i,j)\in A_L^1} (R_{ij} - (\lambda_{ij})_{pilot})u_{ij} + \sum_{(i,j)\in A_L^0} (R_{ij} - (\pi_{ij})_{pilot})u_{ij} + \\ &\sum_{(i,j)\in A_L^1} (\lambda_{ij})_{pilot}(1 - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s) + \sum_{(i,j)\in A_L^0} (\pi_{ij})_{pilot}(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - 0) \\ &\text{subject to } (7)\text{–}(9) \text{ and } (12)\text{–}(14). \end{aligned} \tag{31}$$

From equations (26) and (27) we have $R_{ij} \geq (\lambda_{ij})_{pilot}$ and $R_{ij} \geq (\pi_{ij})_{pilot}$, so $R_{ij} - (\lambda_{ij})_{pilot} \geq 0$ and $R_{ij} - (\pi_{ij})_{pilot} \geq 0$. Model (31) is a minimization problem, so in the optimal solution $\sum_{(i,j)\in A_L^1}(R_{ij} - (\lambda_{ij})_{pilot})u_{ij} = \sum_{(i,j)\in A_L^0}(R_{ij} - (\pi_{ij})_{pilot})u_{ij} = 0$. This completes the proof. □

Remark 4 *We have an equivalent result for the ALD-pilot model (24).*

Proposition 4 *The solution calculated by INT is an optimal solution for model (1)–(14) if the similarity of the pairings between the two problems is 100%.*

Proof. Proof. It suffices to show that the cost of the lower bound obtained by INT is equal to the cost of the upper bound. Let $(\bar{x}_s, \bar{r}_v, \bar{y}_s, \bar{z}_v)$ be an INT solution in which the similarity of the pairings between the two problems is 100%. Hence, $\sum_{(i,j)\in A_L^1}(\lambda_{ij})_{pilot}(1-\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s \bar{y}_s) = 0$ and $\sum_{(i,j)\in A_L^0}(\pi_{ij})_{pilot}$ $(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s \bar{y}_s - 0) = 0$ (or $\sum_{(i,j)\in A_O^1}(\lambda_{ij})_{copilot}(1-\sum_{l\in L}\sum_{s\in S_l} a_{ij}^s \bar{x}_s) = 0$ and $\sum_{(i,j)\in A_O^0}(\pi_{ij})_{copilot}$ $(\sum_{l\in L}\sum_{s\in S_l} a_{ij}^s \bar{x}_s - 0) = 0$), and from Lemma 3 the cost function of INT at the relevant iteration is

$$ALD = \sum_{l\in L}\sum_{s\in S_l} c_s\bar{x}_s + \sum_{l\in L}\sum_{v\in V_l} c_v\bar{r}_v + \sum_{o\in O}\sum_{s\in S_o} c_s\bar{y}_s + \sum_{o\in O}\sum_{v\in V_o} c_v\bar{z}_v. \tag{32}$$

Clearly, $(\bar{x}_s, \bar{r}_v)$ satisfies constraints (2)–(4) and $(\bar{y}_s, \bar{z}_v)$ satisfies constraints (7)–(9). Substituting $(\bar{x}_s, \bar{y}_s)$ into the linking constraints (5)–(6) in model (1)–(14) gives $u_{ij} = 0$. If we put this solution into (1)–(14), the cost is

$$\mathbb{Z} = \sum_{l\in L}\sum_{s\in S_l} c_s\bar{x}_s + \sum_{l\in L}\sum_{v\in V_l} c_v\bar{r}_v + \sum_{o\in O}\sum_{s\in S_o} c_s\bar{y}_s + \sum_{o\in O}\sum_{v\in V_o} c_v\bar{z}_v. \tag{33}$$

Hence, $ALD = \mathbb{Z}$. □

4.5 Integer solution

To obtain integer solutions, the CG is embedded in a branch-and-bound framework. We use two branching heuristics, column fixing and inter-task fixing. In the column fixing strategy, we fix to 1 all the variables with a fractional value greater than a predetermined threshold (0.85 for our tests). If no suitable variable exists, we apply inter-task fixing. For each ordered pair of flights f_1 and f_2, let $r_{f_1 f_2} = \sum_{s\in S(f_1,f_2)} x_s$ be the total flow between the pair. Here $S(f_1, f_2)$ is the subset of schedules covering flight f_2 immediately after f_1. We select the maximum fractional values, $r_{f_1 f_2}$. By setting $r_{f_1 f_2} = 1$ we force these two flights to be covered consecutively in the same schedule. The numerical results show that the solutions produced with this heuristic branching are close to optimality; see Table 6.

5 Computational experiments

We carried out computational experiments on four instances derived from a one-month flight schedule for a major North American airline. The characteristics of the instances are summarized in Table 1. They all contain three crew bases. Each pilot and copilot requests two vacations and a set of PFs each month. The number of pilots and copilots per base is equal and given in Table 2. For example, the last instance contains 5613 flights, 49 airports, 145 pilots and 145 copilots, 290 VRs for pilots and 290 VRs for copilots, and 2500 PFs for pilots and 2500 PFs for copilots. The pairing and schedule feasibility are restricted by the rules stated in Section 2. To compare the performance of INT with the traditional sequential approach (SEQ), we solve each of the instances of Table 1 with both methods. SEQ first solves the crew pairing problem of Zeighami and Soumis (2017) and then the personalized crew assignment problem using CG for a given set of pilots and copilots independently. To define a sequential model for the personalized pilot crew assignment problem, we replace constraint (2) with $\sum_{l\in L}\sum_{s\in S_l} e_p^s x_s = 1 \ \forall p \in P$ by considering constraints (3)–(4) and (10)–(11) and the first two terms of function (1) as the objective function, where P is the set of pairings constructed by the crew pairing problem. The copilot problem is defined similarly. Table 3 reports the number of constraints for each instance for SEQ and INT. For example, for SEQ and the fourth instance, the crew pairing model has 5616 constraints, the pilot assignment model has 1514, and the copilot assignment model has 1514. In INT for the same instance, there are 6051 pilot and copilot constraints, 152187 linking constraints, and 164289 constraints in total.

We conducted our tests on a Linux computer with an Intel Core i7-1770 CPU clocked at 3.40 GHz, using a single processor. Our implementation is coded in C++ using the commercial GENCOL column generation library, version 4.5. The RMPs are solved by CPLEX 12.4. In practice, when CG gets close to the optimal value, new columns with negative reduced costs have little or no effect on the optimal value of the RMP. We therefore stop the CG when the optimal value of the RMP improves by less than 0.01% over 25 iterations for small problems and 15 for large problems. We chose these parameter values based on the results of preliminary tests. Also, we use the improved version of DCA, i.e., MPDCA, as described in Section 4.2. MPDCA is exact when the final phase number k is sufficiently large to ensure the pricing of all feasible columns. Because of the complexity of ALD-copilot and ALD-pilot, we use three phases of DCA ($k = 2$). We also ran a test with $k = 3$, and it gave no significant improvement over $k = 2$.

Table 1: Instance characteristics

		#		No. preferred vacations		No. preferred flights	
Instance	Flights	Airports	Bases	Pilots	Copilots	Pilots	Copilots
1	1013	26	3	66	66	600	600
2	1500	35	3	68	68	750	750
3	1854	41	3	94	94	850	850
4	5613	49	3	290	290	2500	2500

Table 2: Number of pilots and copilots per base

	Base 1		Base 2		Base 3		Total	
Instance	No. Pilots	No. Copilots	No. Pilots	No. Copilots	No. Pilots	No. Copilots	No. Pilots	No. Copilots
1	7	7	20	20	6	6	33	33
2	10	10	9	9	15	15	34	34
3	10	10	30	30	7	7	47	47
4	42	42	78	78	25	25	145	145

Table 3: Number of constraints

	Sequential model			Integrated model			
Instance	Pairing problem	Pilot problem	Copilot problem	Pilot problem	Copilot problem	Linking constraints	Total
1	1016	287	287	1112	1112	32104	34328
2	1503	405	405	1602	1602	35784	38988
3	1857	440	440	1995	1995	40789	44779
4	5616	1514	1514	6051	6051	152187	164289

5.1 Solution process of integrated approach

In this experiment, we study the solution process of INT at each iteration. Table 4 reports the similarity percentage of the pairings for the two problems, the percentage of satisfied VRs and PFs, the CPU time, and the gap at each iteration for each instance.

Using SEQ to solve these problems produces the same pairings for the pilots and copilots. INT produces the same pairings in all instances except the second. The percentage of dissimilarity for the second instance is only 0.35%. The good performance of INT is in part explained by the fact that for all four instances the similarity percentage increases after each iteration while the percentage of satisfied VRs and PFs slightly decreases.

The reported CPU time is the total effort needed to solve each instance at each iteration. For all the instances, most of the CPU time is spent on the first iteration, and this can be explained as follows. First, choosing a good initial solution has a positive influence on the performance of DCA; it can result in fewer CG iterations and hence a lower CPU time. We warm-start the pilot and copilot problems after the first iteration. After solving each problem we use the current optimal pairings as the initial solution for the other problem; see Section 4.3. Second, using good initial columns reduces the CPU time of CG. We use the optimal solution of the current iteration of ALD-pilot as the initial solution for the next iteration of this problem. When adding initial columns to the master problem of CG, we take into account that some of these columns will subsequently be removed by DCA if they are not compatible with the pairings found by ALD-copilot. They can be re-introduced when the initial partition is reconsidered by DCA. The same strategy applies to ALD-copilot. In particular, the computational times are reduced considerably after the first iteration. The CPU times are typically divided by a factor of three on each instance.

The gap corresponds to the relative difference between the values of the lower bound and upper bound obtained by INT for the global objective. The results show that the percentage gaps are between 0.00% and 0.01%, and this justifies the use of INT.

INT reaches a (nearly) optimal solution in three or four iterations for the large and complex problem. For example, in the third instance, the similarity is 91.35% at the first iteration and 100% at the end of the fourth iteration. The percentage of satisfied VRs is unchanged for the pilots and 0.04% lower for the copilots. The percentage of satisfied PFs is 0.15% lower for the pilots and 0.43% lower for the copilots. The global optimality gap is 1.52% at the first iteration and 0% at the last iteration.

Table 4: Solution process of INT

			Pilots		Copilots			
Instance	Iteration	Similarity percentage	Satisfied VRs (%)	Satisfied PFs (%)	Satisfied VRs (%)	Satisfied PFs (%)	CPU (min)	Gap (%)
	WT	82.14	96.96	37.66	98.48	39.00	3.37	-
	1	92.44	96.96	37.66	98.48	38.66	3.55	1.45
1	2	97.61	96.96	37.16	98.48	38.50	1.24	0.20
	3	100.00	96.96	37.00	98.48	38.00	1.05	0.00
	WT	83.49	92.64	43.65	97.05	41.73	12.02	-
	1	89.50	92.64	43.65	97.05	41.50	12.15	1.75
2	2	95.22	92.64	43.55	97.05	41.46	4.06	0.45
	3	98.00	92.64	43.33	97.05	41.46	3.52	0.12
	4	99.65	92.46	43.33	97.05	41.06	3.41	0.01
	WT	81.75	84.00	50.75	82.97	51.17	26.23	-
	1	91.35	84.00	50.75	81.95	50.90	26.40	1.52
3	2	97.43	84.00	50.72	81.95	50.88	7.17	0.18
	3	99.27	84.00	50.70	81.95	50.88	7.26	0.05
	4	100.00	84.00	50.70	81.91	50.47	6.55	0.00
	WT	80.15	98.62	50.12	98.62	50.68	1206.45	-
	1	96.47	98.62	50.00	98.62	50.56	1224.33	0.22
4	2	99.96	98.27	49.40	98.62	50.28	175.23	0.03
	3	100.00	98.27	49.40	98.62	50.28	150.38	0.00

5.2 Impact of Lagrangian multipliers on the similarity of pairings

To highlight the impact of the Lagrangian multipliers on the similarity percentage, we solve each problem using CG and DCA without transferring these values (labeled WT in Table 4). The results clearly show that there is a significant reduction in the similarity percentage when we ignore the Lagrangian multiplier values. For example, for the first iteration of the largest instance, the similarity percentage is 80.15% without the Lagrangian values and 96.47% at the first iteration with them. That is an improvement of 16.32% at the first iteration while the percentage of satisfied PFs and VRs decreases by less than 1%.

5.3 Computational time, iterations, and cost for integrated and sequential approaches

In Table 5, we compare the total CPU time (in minutes) to solve each of the four instances with the two approaches. For SEQ, the total CPU time consists of the time to solve the pairing problem and the time to solve the pilot and copilot assignment problems. For INT, the total CPU time consists of the time to compute the initial partition, i.e., the time to solve the crew pairing problem of Zeighami and Soumis (2017), and the time to solve ALD-pilot and ALD-copilot. We report the total solution cost obtained with both approaches. This includes the cost of the pairings, bonuses for satisfied PFs, penalties for unsatisfied VRs, and the cost associated with the similarity percentage. The last two columns of Table 5 give the ratios of the CPU times of INT and SEQ and the percentage cost savings achieved by the INT solutions. These results show that INT can yield significant cost savings: between 12.76% and 24.20%, with an average of 18.89%. Just a few iterations suffice to find a good solution, and the computational time is close to that for SEQ: 1.65 times longer on average.

Table 5: Computational Time and Cost for Integrated and Sequential Approaches

Instance	SEQ approach		INT approach				
	CPU (min)	Cost	CPU (min)	Cost	Number of iterations	CPU INT/SEQ	Cost saving by INT (%)
1	5.47	1551144	11.42	1250844	3	2.08	19.35
2	19.56	1903930	36.28	1660930	4	1.85	12.76
3	61.50	2442300	80.15	2047500	4	1.30	19.28
4	1393.58	3672890	1943.52	2783900	3	1.39	24.20
Average						1.65	18.89

5.4 Computational gap in integrated and sequential approaches

In this experiment, we study the optimality gap of each instance. The gap is the percentage difference between the best lower bound and the best integer solution obtained by CG. To find a lower bound, we stop CG at each branching node if the optimal value of the RMP improves by less than 0.01% over 25 iterations for small problems and 15 for large problems. The lower bound improves during the branch and bound process since we explore many nodes. To obtain integer solutions, we use the branching heuristics discussed in Section 4.5. Table 6 gives the results of this experiment. For SEQ, three optimality gaps are reported: the crew pairing gap corresponds to the crew pairing problem, and the pilot and copilot gaps correspond to the pilot and copilot assignment problems. For INT, two optimality gaps are reported, for ALD-pilot and ALD-copilot. The percentage gaps are small, indicating that the two approaches produce good solutions. On average the percentage gap for SEQ varies between 0.18 and 0.47, and that for INT varies between 0.03 and 0.10.

Table 6: Optimality gap (%) for integrated and sequential approaches

Instance	SEQ approach			INT approach	
	Pairing problem	Pilot problem	Copilot problem	ALR-pilot problem	ALR-copilot problem
1	0.07	0.10	0.00	0.02	0.00
2	1.86	0.00	0.22	0.06	0.05
3	0.88	0.55	0.21	0.00	0.00
4	0.10	0.38	0.30	0.06	0.35
Average	0.47	0.25	0.18	0.03	0.10

5.5 Comparisons of pairing costs and coverage of VRs and PFs

In this experiment, we study the performance of INT and SEQ in terms of pairing costs and the percentage of satisfied VRs and PFs. The objective function (1) finds a trade-off between maximizing the number of satisfied VRs and PFs and minimizing the total cost of the pairings and their dissimilarity. Table 7 reports the total pairing cost obtained by the two approaches. On average, INT increases the pairing cost by only 0.55% above that of SEQ. This increase might be necessary to increase the number of satisfied VRs and PFs. However, to force the model to stay close to the SEQ pairing cost, we can add the constraints $\sum_{l\in L}\sum_{s\in S_l}(\sum_{p\in P_s} c_p)x_s \leq (1+\epsilon)N$ and $\sum_{o\in O}\sum_{s\in S_o}(\sum_{p\in P_s} c_p)y_s \leq (1+\epsilon)N$ in model (1)–(14), where N is the pairing cost obtained by SEQ. Tables 8 and 9 give the results: INT performs significantly better than SEQ. On average, INT increases the coverage of VRs by 43.50% for the pilots and 38.49% for the copilots. On average, it increases the coverage of PFs by 14.85% for the pilots and 12.67% for the copilots. These improvements can be explained by the fact that SEQ does not take into account the crew assignment constraints and objective during the construction of the pairings. The pairings generated by the pairing problem may not be suitable for the objective of the assignment problem, so it is difficult to maximize the satisfaction of VRs and PFs in SEQ.

Remark 5 *A (co)pilot may request flights that have conflicts in the departure or arrival times. Therefore, it is difficult to satisfy a high percentage of the PFs.*

Table 7: Comparisons of pairing cost

Instance	SEQ approach	INT approach	Improvement by INT
1	1274922	1280044	- 0.40
2	1678746	1688880	- 0.60
3	1952742	1967500	- 0.75
4	2966890	2980900	- 0.47
Average			- 0.55

Table 8: Comparisons of satisfied VRs (%)

	SEQ approach		INT approach		Improvement by INT (%)	
Instance	Pilot problem	Copilot problem	Pilot problem	Copilot problem	Pilot problem	Copilot problem
1	45.00	60.00	98.00	98.00	53.00	38.00
2	42.00	52.00	92.00	97.00	50.00	45.00
3	42.00	39.00	84.00	81.00	42.00	42.00
4	68.96	69.65	98.26	98.26	29.30	28.97
Average					43.50	38.49

Table 9: Comparisons of satisfied PFs (%)

	SEQ approach		INT approach		Improvement by INT (%)	
Instance	Pilot problem	Copilot problem	Pilot problem	Copilot problem	Pilot problem	Copilot problem
1	26.00	28.00	37.00	39.00	11.00	11.00
2	28.00	30.00	43.00	41.00	15.00	11.00
3	32.00	34.00	52.00	50.00	20.00	16.00
4	36.00	37.60	49.40	50.28	13.40	12.68
Average					14.85	12.67

6 Conclusion

We have introduced an integrated crew pairing and personalized crew assignment model to generate personalized monthly schedules for a given set of pilots and copilots simultaneously. Each pilot and copilot requests a set of preferred flights and vacations each month. Our model keeps the pairings in the two problems as similar as possible to reduce the propagation of perturbations. We have presented an integrated approach based on ALD, CG, and DCA. This novel approach uses Lagrangian decomposition in an alternating fashion. It introduces a new way to update the Lagrangian multipliers by estimating the effect of decisions made in one problem on the other. We proved that our approach finds an optimal solution when the sets of pairings are identical. We studied real-world instances from a major US carrier, and the results show that a few iterations suffice to find a (nearly) optimal solution for the large and complex problem. Also, the integrated approach yields significant improvements over the sequential approach in terms of satisfied VRs and PFs.

References

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. Operations Research, 46(3):316–329, 1998.

C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. Transportation Science, 37(4):368–391, 2003.

V. Cacchiani and J.-J. Salazar-González. Optimal solutions to a real-world integrated airline scheduling problem. Transportation Science, 51(1):250–268, 2016.

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

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

M. Desrochers and F. Soumis. A generalized permanent labelling algorithm for the shortest path problem with time windows. INFOR: Information Systems and Operational Research, 26(3):191–212, 1988.

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

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

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

M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. Mathematical Programming, 39(2):215–228, 1987.

Y. Guo, T. Mellouli, L. Suhl, and M. P. Thiel. 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, 2006.

A. Kasirzadeh. Optimisation intégrée des rotations et des blocs mensuels personnalisés des équipages en transport aérien. Ph.D. thesis, Department of Mathematics and Industrial Engineering, Polytechnique Montréal, 2015.

A. Kasirzadeh, M. Saddoune, and F. Soumis. Airline crew scheduling: models, algorithms, and data sets. EURO Journal on Transportation and Logistics, 6(2):111–137, 2017.

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

A. Mercier and F. Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. Computers & Operations Research, 34(8):2251–2265, 2007.

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

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

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

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

S. Shao, H. D. Sherali, and M. Haouari. A novel model and decomposition approach for the integrated airline fleet assignment, aircraft routing, and crew pairing problem. Transportation Science, 2015.

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

V. Zeighami and F. Soumis. Combining Benders decomposition and column generation for integrated crew pairing and personalized crew assignment problems. Technical Report G–2017–41, Cahiers du GERAD:1–22, 2017.