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

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 simultaneous for pilots and copilots

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
Les Cahiers du GERAD
G–2018–37

Abstract: The airline crew scheduling problem consists of determining crew 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 size and complexity, this problem is usually solved sequentially in two main steps: the crew pairing followed by the crew assignment. However, finding a global optimal solution via sequential approach may become impossible because the decision domain of the crew assignment problem is reduced by previously made decisions in the crew pairing problem. In this paper, we consider the pilot and copilot crew scheduling problems in a personalized context where each pilot and copilot requests a set of preferences flights and vacations per 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 where we keep the pairings in the two problems as similar as possible so the propagation of the perturbations during the operation is reduced. To solve this model, we develop an integrated approach based on combining alternating Lagrangian decomposition, column generation, and dynamic constraint aggregation. We conduct computational experiments on a set of real instances from a major US carrier. The integrated approach produced significant cost savings and satisfaction of crew preferences in comparison with the traditional sequential approach commonly used in practice.

Keywords: Airline crew scheduling, column generation, dynamic constraint aggregation, alternating Lagrangian decomposition, arc-reduced costs

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 crew scheduling problem is a complex task for most airlines. Since the crew cost is the largest after the fuel cost, this problem is one of the most important problems in the airline planning. Given a set of flights are defined by departure/arrival stations and fixed departure/arrival dates and times to be operated by the same aircraft fleet, the goal of crew scheduling problem is to construct pairings from these flights and assigning them to a set of available crew members to construct their individual schedules. When the number of flights increases, this problem is usually solved in two steps: crew pairing problem followed by crew assignment problem.

A pairing is a sequence of flights, connections, and rests starting and ending at the same crew base. The crew pairing problem consists of determining a set of feasible pairings given the scheduled flights 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 used for this problem: the bidline approach started to be used in some North American airlines. This produces anonymous monthly schedules, and the crew members select their schedules 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. In the personalized approach with a global objective, which is often employed by European airlines and has recently started to be used by American airlines, all the monthly schedules are produced simultaneously to maximize an objective function. This may be the sum of the personal satisfactions without any advantages for the most seniority.

However, using the sequential approach to solve crew scheduling problem 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 building 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 crew preferences in the crew assignment stage and the global solution of the crew scheduling problem is not guaranteed.

During the last decades, different types of models and methods have been introduced for the crew scheduling problem. We refer to the recent surveys by Kasirzadeh et al. (2014) and Eltoukhy et al. (2017). While solving the crew scheduling problem sequentially may lead to poor solutions, only a few researchers have investigated the integrated models and solutions. Cordeau et al. (2001) and Mercier et al. (2005) proposed a solution methodology based on column generation (CG) and Benders decomposition for integrated aircraft routing and crew pairing. Zeghal and Minoux (2006) propose an integer linear programming model using clique constraints for integrated pairing and bidline bloc. They consider small problems with 59 to 210 flights and a few flights per pairing. Sandhu and Klabjan (2007) introduces a model that completely integrates the fleeting and crew pairing stages. They proposed two solution methodologies based on a combination of Lagrangian relaxation and column generation and the other one is a Benders decomposition approach. Integrated aircraft routing, crew scheduling, and flight retiming were studied by Mercier and Soumis (2007). Souai and Teghem (2009) described a genetic algorithm for integrated pairing and personalized bloc. Papadakos (2009) introduces a set covering model for integrated fleet assignment, maintenance routing, and crew pairing and a method based on Benders decomposition combined with CG. Saddoune et al. (2011, 2012) develop 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. They consider two weekly instances with 898 and 960 flights. Shao et al. (2015) considered the integration of the fleet assignment, aircraft routing, and crew pairing problems. Kasirzadeh (2015) presented a heuristic algorithm based on CG and DCA for integrated crew pairing and personalized assignment. Cacchiani and Salazar-González (2016) proposed two mixed integer linear programming models for integrated fleet assignment, aircraft routing, and crew pairing. Zeighami and Soumis (2017) proposed an integrated model for the crew pairing and personalized assignment problems for a given set of pilots and copilots. They developed a solution methodology based on Benders decomposition and CG.

In this paper, we consider the integrated pilot and copilot crew pairing and personalized crew assignment problems in a personalized context with a global objective where each pairing requires one pilot and one copilot, and each (co)pilot has a set of Preferences Flights (PFs) and Vacations Request (VRs) per month. The pilots and copilots have different schedule to maximizing their preferences, however, to reduce the propagation of the perturbations during the operation, they must have similar pairings as much as possible. To this end, we optimize the schedules for the pilots and copilots simultaneously, taking their preferences into account; see Figure 1.

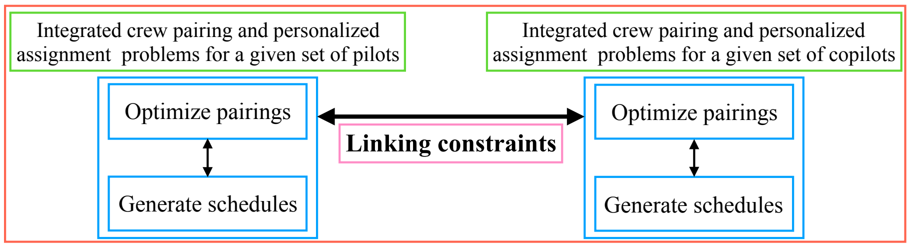

Figure 1: Integrated crew pairing and personalized assignment problems simultaneous for pilots and copilots within a comprehensively integrated framework

In summary, this paper makes the following contributions. We introduce a novel integrated crew pairing and personalized assignment model to generate personalized monthly schedules in a single step for a given set of pilots and copilots simultaneously where we keep the pairings in the two problems as similar as possible. To solve this integrated problem, we develop a solution methodology based on alternating Lagrangian decomposition (ALD), CG and DCA, which has not yet been investigated in the literature. The developed approach use Lagrangian decomposition in an alternating fashion proposing a new way to update Lagrangian multipliers. The solution process iterates between the integrated pilot model and the integrated copilot model by estimating the effects of decisions in the first problem on the second problem.

The paper is organized as follows: Section 2 gives the problem statement. The mathematical formulation is given in Section 3. The solution methodology is outlined in Section 4 and in Section 5, we describe our experimental results and observations. Finally, Section 6 presents conclusions along with possible directions for future development.

2 Problem statement

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

Each flight is defined by the specific 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 rest 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 between 4 and 8 hours in a duty, 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 Kasirzadeh et al. (2014).

(Co)Pilots are trained for one specific aircraft fleet and they are associated with a crew base. All pairings assigned to a (co)pilot must start and end at his base. A base is a large airport where pilots and copilots are stationed. We assume that there is a fixed number of (co)pilots at each base. In our test, each (co)pilot requests a set of PFs and VRs per month. A schedule is a sequence of pairings separated by rest periods. We build monthly schedules for the (co)pilots that cover all the pairings previously built. 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 times between any two consecutive pairings is 8. A penalty cost is associated with each unsatisfied VR and each uncovered pairing. The objective function minimizes the number of unsatisfied VRs and maximizes the number of satisfied PFs as much as possible.

3 Mathematical formulation

In this section, we propose a novel integrated crew pairing and personalized crew assignment for a given set of pilots and copilots within a completely integrated framework. The integrated model constructs the pairings and monthly schedules for each pilot and copilot in a single step directly from flights instead of pairings, where we keep the pairings in the two problems as similar as possible, considering the feasibility rules impacting the pairings and monthly schedules. The proposed model can be formulated using the following notations. We define the set of pilots by L and the set of copilots by 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 be 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 chosen by pilot $l \in L$ and 0 otherwise. The binary variable y_s is equal to 1 if schedule $y \in S_o$ is chosen by copilot $o \in O$ and 0 otherwise. The binary variable r_v is 1 if vacation $v \in V_l$ is not covered by pilot $l \in L$ and 0 otherwise. The binary variable z_v is 1 if vacation $v \in V_o$ is not covered by copilot $o \in O$ and 0 otherwise. Let c_v be the penalty costs for uncovered vacation $v \in V_l$ (or $v \in V_o$). The binary constant b_v^s is equal to 1 if $v \in V_l$ (or $v \in V_o$) is covered by schedule $s \in S_l$ (or $s \in S_o$). Let n_s be the number of preferred flights in scheduled $s \in S_l$ (or $s \in S_o$) and B denote the bonus cost (a negative cost) for covering each preferred flight. 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 that covered by schedule s. We denote by c_p the cost of pairing p. Let A be the all feasible connection arcs between two successive flight legs. For every connection arc $(i,j) \in A$ and every schedule s, we define the binary constant a_{ij}^s equal to 1 if connection arc ij between flight i and j is covered by schedule s. The binary variable e_{ij} is 1 if connection arc ij is covered just only by pilot problem or copilot problem and 0 otherwise. Let R_{ij} be the penalty cost for connection arc ij if it is equal to 1.

Using this notation, the propose integrated model is as follow:

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

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, PFs and minimizing the total cost of the pairings and dissimilarity of pairings between two problems. Set-partitioning constraints (2) and (7) ensure that each flight is covered by exactly one pilot and one copilot. Constraints (3) and (8) ensure that each vacation is covered by one (co)pilot or not covered. Linking constraints (5) and (6) offers the possibility of limiting the total number of different pairings in two problems. The integrality conditions are defined by constraints (10)–(14).

4 Solution methodology

In this section, we provide the layout of our algorithm. The proposed integrated model (1)–(14) contains a large number of linking constraints. To handle these linking constraints, a solution approach based on Lagrangian decomposition is proposed in an alternating fashion. The solution process iterates between two Lagrangian subproblems. The first subproblem is the integrated pilot problem and the second subproblem is the integrated copilot problem. In both problems the number of variables grows exponentially as the number of flights increases. We avoid this issue by using a CG method. However, due to the high degeneracy, CG becomes inefficient when the number of set partitioning constraints is large and the columns are dense (more than 8–12 nonzero elements; see Elhallaoui et al. (2005)). In our problem, the number of nonzeros varies between 30 and 45. To overcome this difficulty and accelerate the solution process, we combine CG with DCA. A heuristic branch-and-bound method is used to compute integer solutions. First, we explain the functioning of this approach and then we describe how the model (1)–(14) can be decomposed through Lagrangian decomposition.

4.1 Column generation

CG (Desaulniers et al. (2005)) is an iterative method to solve large-scale problems especially when there are a large number of variables. CG decomposes the main problem into a restricted master problem (RMP) and several pricing problems. The restricted master problem is a restriction of the master problem and contains only a subset of the columns. New variables for the master problem are then generated at each iteration by solving pricing problems, expecting to improve the RMP objective function. At each iteration of the CG process, 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. CG iterate until no negative reduced cost variables are identified. In this case, the current RMP primal solution is also optimal for the master problem. In practice, when CG getting close to the optimal value, new columns with negative reduced cost have little or no effect on the optimal value of the RMP. To avoid this situation, we stop the CG when the optimal value of the RMP improves by less than 0.1% over five iterations.

4.1.1 Pricing problems

In this Sections, we describe the pricing problems. The pricing problems are defined on a directed acyclic time-space network. Each pricing problem corresponds to a resource-constrained shortest path problem that is solved using a label-setting algorithm (Irnich and Desaulniers 2005).

There is one pricing problem for each (co)pilot. Each pricing problem contains 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 to start a duty there is a waiting node for each departing node. The midnight nodes are used to define the start and end of each midnight. The network 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*. 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. Finally, 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 three 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, and maximum total duty duration, a minimum number of days off, maximum number of consecutive working days, and maximum credited flying time.

4.1.2 Pricing problem solver

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. 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 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 two 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

The theoretical framework for DCA was introduced by Vance et al. (1997), and Elhallaoui et al. (2005) provided the first implementation. DCA is a new version of the standard column generation method in which deals with reducing the number of set-partitioning constraints in the restricted master problem (RMP) by aggregating some of them. This aggregation is performed by aggregating clusters of the RMP set-partitioning constraints and keeping 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. During the solution process of column generation, this constraint aggregation can change dynamically until an optimal solution is found. A column is said to be compatible with partition Q 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 cannot 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, a dual variable disaggregation procedure based on shortest-path calculations. An advanced 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 a phase number 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. The ARMP is solved 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 no negative-reduced-cost column, the algorithm increment k and go to the next phase $(k+1)$ or stop 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 all or a subset of the compatible columns to the ARMP, and the MPDCA moves to its next iteration. Otherwise, a test is performed to determine whether the current partition should be changed. The partition doesn't need to change if the reduced cost of the least reduced-cost compatible column is less than the reduced cost of the least reduced-cost incompatible column times a predetermined multiplier, otherwise, the partition is updated. An aggregation is performed according to a partition Q of the flights in F into clusters. A cluster is a pairing, and a set of pairings that covers all flights is a partition. To apply MPDCA for solving the integrated model (1)–(14), we must define an initial partition Q. The initial partition is built from the solution of Zeighami and Soumis (2017). Each column corresponds to a feasible monthly schedule for a pilot (or copilot). MPDCA is exact if the last phase number k is sufficiently large to ensure the pricing of all feasible columns. Given the high complexity of the integrated model (1)–(14), we use three phase, namely, $k = 0, k = 1$ and $k = 2$ for our experiments. Solving the pricing problems in low number phases is computationally easier.

4.3 Alternating Lagrangian decomposition

To reduce the complexity of the solution process, model (1)–(14) can be decomposed into two smaller subproblems via Lagrangian decomposition, solvable by combining CG and DCA. First Lagrangian subproblem, called ALR-pilot, represents the integrated pilot personalized assignment problem. The second subproblem, called ALR-copilot, represents the integrated copilot personalized assignment problem. The classical Lagrangian decomposition via subgradient or bundle methods usually needs many iterations to reach an optimal solution. To overcome this weakness, we use Lagrangian decomposition in an alternating fashion proposing a new way to update Lagrangian multipliers. The solution process iterates between ALR-pilot and ALR-copilot by estimating the effect of decisions in the first problem on the second problem and vice versa.

We first solve the integrated pilot model (2)–(4) considering the two first terms of function (1) as the objective function and let $\bar{x}_s$ $(s \in S_l; l \in L)$ be the optimal solution. Let A_l^1 be the set of connection arcs that are covered by pilot problem (i.e., $\sum_{l \in L} \sum_{s \in S_l} a_{ij}^s \bar{x}_s = 1$), and A_l^0 be set of connection arcs that are not covered by pilot problem (i.e., $\sum_{l \in L} \sum_{s \in S_l} a_{ij}^s \bar{x}_s = 0$). Where, $A = A_l^1 \cup A_l^0$. Hence, the resulting linking constraints (5)–(6) are

$$1 - \sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s \leq u_{ij} \qquad \forall (i,j) \in A_l^1 \subseteq A \tag{15}$$

$$\sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s - 1 \leq u_{ij} \qquad \forall (i,j) \in A_l^1 \subseteq A \tag{16}$$

$$0 - \sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s \leq u_{ij} \qquad \forall (i,j) \in A_l^0 \subseteq A \tag{17}$$

$$\sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s - 0 \leq u_{ij} \qquad \forall (i,j) \in A_l^0 \subseteq A \tag{18}$$

Clearly constraints (16) and (17) are redundant. Let $\mu = (\mu_{ij} \geq 0 | (i,j) \in A)$ and $\pi = (\pi_{ij} \geq 0 | (i,j) \in A)$ be multipliers associated to constraints (15) and (18), respectively. The Lagrangian decomposition is obtained by dualizing these linking constraints. The resulting Lagrangian subproblem for copilot problem is

$$\text{ALR-copilot } (\mu, \pi) = \min \sum_{o \in O} \sum_{s \in S_o} c_s y_s + \sum_{o \in O} \sum_{v \in V_o} c_v z_v + \sum_{(i,j) \in A} R_{ij} u_{ij} + \sum_{(i,j) \in A_l^1} \mu_{ij} (1 - \sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s - u_{ij}) + \sum_{(i,j) \in A_l^0} \pi_{ij} (\sum_{o \in O} \sum_{s \in S_o} a_{ij}^s y_s - 0 - u_{ij}) \quad (19)$$

subject to (7)–(9) and (12)–(14).

For ALR-pilot we have the same argument as ALR-copilot,

$$\text{ALR-pilot } (\mu, \pi) = \min \sum_{l \in L} \sum_{s \in S_l} c_s x_s + \sum_{l \in L} \sum_{v \in V_l} c_v r_v + \sum_{(i,j) \in A} R_{ij} u_{ij} + \sum_{(i,j) \in A_o^1} \mu_{ij} (1 - \sum_{l \in L} \sum_{s \in S_l} a_{ij}^s x_s - u_{ij}) + \sum_{(i,j) \in A_o^0} \pi_{ij} (\sum_{l \in L} \sum_{s \in S_l} a_{ij}^s x_s - 0 - u_{ij}) \quad (20)$$

subject to (2)–(4) and (10)–(12).

Flowchart of INT approach is given in Figure 2. INT approach starts from a set of initial pairings as an initial solution. In the first iteration, we solve ALR-pilot using CG and DCA and construct personalized monthly schedules for the pilots. The set of initial pairings is updated accordingly, and we update Lagrangian multipliers, as explained in Section 4.3.1. Then, before starting to solve ALR-copilot, the calculated Lagrangian multipliers from ALR-pilot must be projected on the corresponding arcs in ALR-copilot's pricing problems, and new pairings into account as an initial solution, we solve ALR-copilot using CG and DCA and construct personalized monthly schedules for the copilots. This process continues until the relative difference between the lower and upper bounds is less than or equal to a predetermined threshold.

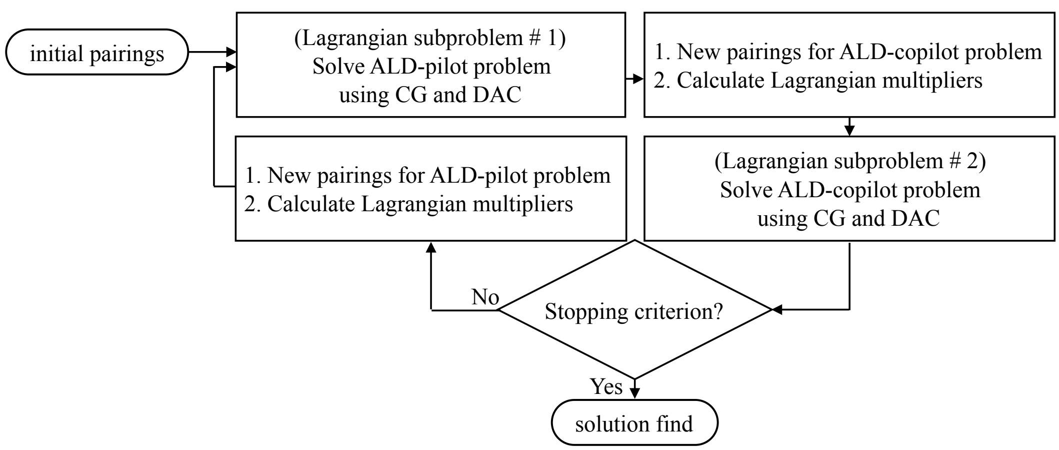

Figure 2: Flowchart of INT approach

4.3.1 Computing Lagrangian multipliers

To compute Lagrangian multipliers, we introduce the concept of arc reduce cost so as to take the benefits of more information between two problems. In a traditional CG, the arc-reduced costs are not directly available and require additional information from the pricing problem to compute them. It is defined as:

$$\bar{\bar{c}}_{ij} = c_{ij} - \alpha_{ij} + (\Pi_j - \Pi_i) \quad (21)$$

where Π_i and Π_j are the minimum cost from origin to the node i and j in the pricing problem respectively, α_{ij} is the dual value of arc ij, and c_{ij} is the cost of arc ij. We calculate minimum label cost on each departure and arrival node by using labeling algorithm. Let us start with ALR-pilot. For the sake of more clarity, the following example is given as shown in Figure 3 (for clarity reasons, some arcs and labels are truncated or omitted). Assume that we have three pilots: pilot A, B and C. Without loss generality, we assume that in the optimal solution of ALR-pilot, pilot A is covered flights i, j and k. Pilot B is covered flights m, n, o and p. Finally, pilot C is covered flights q, s and r. We denote minimum label cost on each node by Π. For example notations $\bar{\Pi}_i^A$ and Π_i^A indicate the minimum label cost on departure and arrival flight i that obtained by pilot A respectively. Now before starting to solve each problem, We calculate arc reduce cost on

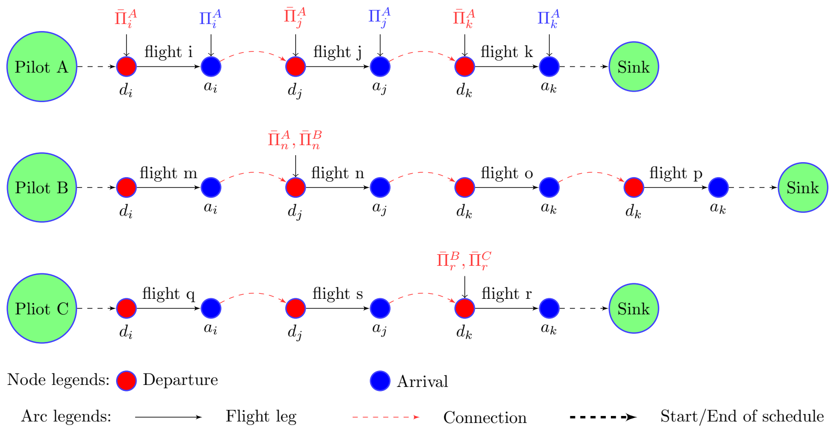

Figure 3: Finding minimum label cost on each departure and arrival node for each pilot

each connection arc in the pilot's pricing problems (see Figure 4). Related to each pilot (or copilot) there is one pricing problem, hence, connection arc ij can exist in more than one pricing problem, therefore, there is more than one possibility for calculate arc reduce cost on connection arc ij. To overcome this difficulty, we use the pilot how covered first flight and the pilot how covered second flight related to this connection arc in the optimal solution.

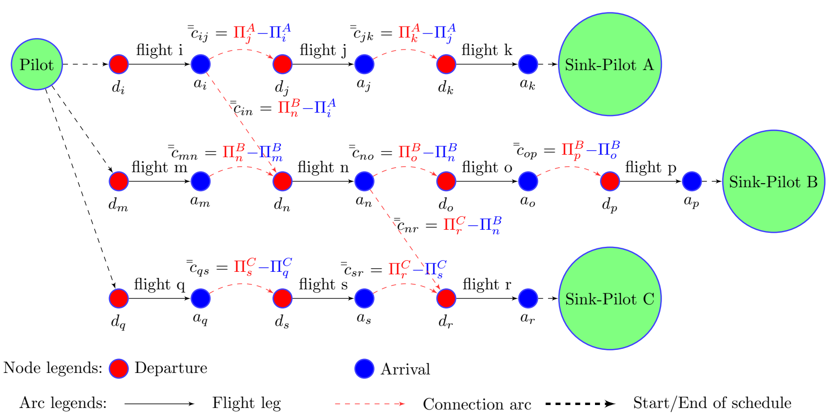

Figure 4: Calculate arc reduce cost on each connection arc in the pilot problem

The Lagrangian multipliers are calculated using the following formula:

$$\mu_{ij} = max\{min\{\bar{\bar{c}}_{ij}, R_{ij}\}, 0\} \qquad \forall (i,j) \in A_l^1 \tag{22}$$

$$\pi_{ij} = max\{min\{\bar{\bar{c}}_{ij}, R_{ij}\}, 0\} \qquad \forall (i,j) \in A_l^0 \tag{23}$$

Before starting to solve copilot problem, the calculated multipliers on each arc in ALR-pilot must be projected on the corresponding arcs in the copilot's pricing problems, then solve copilot problem (see Figure 5). If connection arc ij is covered in ALR-pilot, we add value $-\mu_{ij}$ on connection arc ij in the all ALR-copilot's pricing problems that include this arc, but if connection arc ij is not covered, we add value $+\pi_{ij}$ on connection arc ij; see Figure 5.

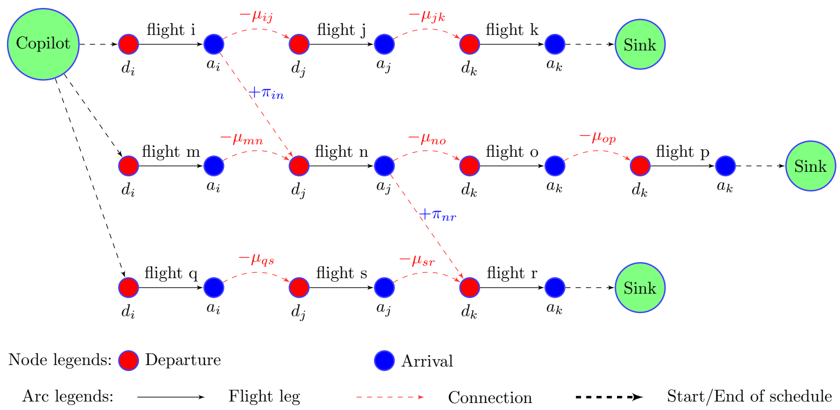

Figure 5: Projected Lagrangian multipliers' pilot problem on the corresponding arcs in the copilot's pricing problems

Remark 1 *Clearly model (19) is equivalent with following model*

$$ALR\text{-}copilot\ (\mu,\pi) = min \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v + \sum_{(i,j)\in A^1_l}(R_{ij}-\mu_{ij})u_{ij} + \sum_{(i,j)\in A^0_l}(R_{ij}-\pi_{ij})u_{ij} +$$
$$\sum_{(i,j)\in A^1_l}\mu_{ij}(1-\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s) + \sum_{(i,j)\in A^0_l}\pi_{ij}(\sum_{o\in O}\sum_{s\in S_o} a^s_{ij}y_s - 0)$$
$$\text{subject to (7)–(9) and (12)–(14).}$$

By the Equations (22) and (23) $R_{ij} \ge \mu_{ij}$ *and* $R_{ij} \ge \pi_{ij}$, *hence, in optimality* $u_{ij} = 0$. *Therefore,* $\sum_{(i,j)\in A^1_l}(R_{ij}-\mu_{ij})u_{ij} = 0$ *and* $\sum_{(i,j)\in A^0_l}(R_{ij}-\pi_{ij})u_{ij} = 0$. *For the model (20) the argument is similar.*

Definition 1 *Let* $P(\bar{y}_s,\bar{z}_v)_{ALD-copilot}$ *and* $P(\bar{x}_s,\bar{r}_v)_{ALD-pilot}$ *indicate the cost of ALD-copilot and ALD-pilot with consider just two first terms of objective function (19) and (20), respectively. Let* $L_{\bar{x}_s}(\mu,\pi)_{ALD-copilot}$ *be the cost of the Lagrangian term that obtained by the solution of ALD-pilot problem and respect to Lagrangian multipliers from ALD-copilot problem. Let* $L(\mu,\pi) = P(\bar{x}_s,\bar{r}_v)_{ALD-pilot} + P(\bar{y}_s,\bar{z}_v)_{ALD-copilot} + L_{\bar{x}_s}(\mu,\pi)_{ALD-copilot}$.

Lemma 1 $L(\mu,\pi)$ *is a lower bound on the optimal objective function value of the integrated model (1)–(14).*

Proof. Clearly $P(\bar{y}_s,\bar{z}_v)_{ALD-copilot}$ (blue point) is a lower bound on the copilot problem and $P(\bar{x}_s,\bar{r}_v)_{ALD-pilot}+ L_{\bar{x}_s}(\mu,\pi)_{ALD-copilot}$ (red point) is a approximation for the pilot problem. □

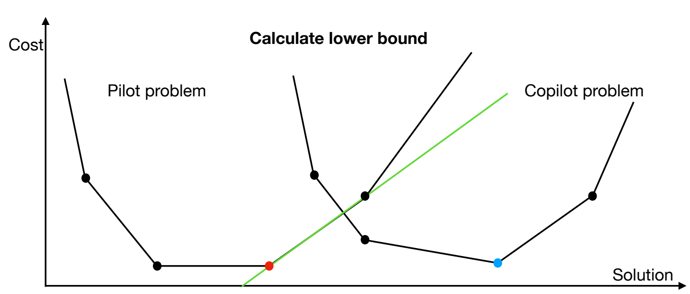

Figure 6: Calculate lower bound

Lemma 2 *During the solution process of ALD, if the percentage of similarity of pairings between two problems be 100%, the current solution is a feasible and optimal solution for the integrated model (1)–(14).*

Proof. It suffices to show that the cost of lower bound that obtained by ALD is equal to the cost of upper bound. Let the percentage of similarity of pairings between two problems be 100%. It is mean that

$\sum_{(i,j)\in A_l^1} \mu_{ij}(1 - \sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s) = 0$ and $\sum_{(i,j)\in A_l^0} \pi_{ij}(\sum_{o\in O}\sum_{s\in S_o} a_{ij}^s y_s - 0) = 0$. Also from Remark 1, we have $u_{ij} = 0$. Hence, the optimal value of ALD in the relevant iteration is

$$ALD(\mu, \pi) = \sum_{l\in L}\sum_{s\in S_l} c_s x_s + \sum_{l\in L}\sum_{v\in V_l} c_v r_v + \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v \tag{24}$$

Clearly this solution satisfy linking constraints (5)–(6) and is a feasible solution for integrated model (1)–(14). If we put this solution into the integrated model, the cost is

$$\mathbb{Z} = \sum_{l\in L}\sum_{s\in S_l} c_s x_s + \sum_{l\in L}\sum_{v\in V_l} c_v r_v + \sum_{o\in O}\sum_{s\in S_o} c_s y_s + \sum_{o\in O}\sum_{v\in V_o} c_v z_v \tag{25}$$

Hence, $ALD(\mu, \pi) = \mathbb{Z}$. □

4.4 Integer solution

The linear relaxation solutions computed by column generation may be fractional. To obtain integer solutions, the column generation algorithm is embedded in a branch-and-bound framework. We use two branching heuristics, column fixing and inter-task fixing. In 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 these two flights. Where $S(f_1, f_2)$ is the subset of schedules covering flights f_2 immediately after pairing f_1. We select the maximum fractional values, $r_{f_1 f_2}$. By setting $r_{f_1 f_2} = 1$ we impose these two flights to be covered consecutively in the same schedule respectively.

5 Computational experiments

In this section, we present computational experiments that were carried out on three instances derived from a one-month flight schedule of a major North American airline. The characteristics of the instances are summarized in Table 1. They all contain three crew bases. Each (co)pilot requests two vacations and a set of preferences flights per month. The numbers of pilots and copilots per base are equal and given in Table 2. For example, the first instance contains 1013 flights, 26 airports, 33 pilots and 33 copilots, a total amount of 66 requested vacations for pilots and 66 requested vacations for copilots, a total amount of 600 preferences flights for pilots and 600 preferences flights for copilots. To compare the performance of proposed integrated (INT) approach with the traditional sequential (SEQ) approach, we solve each of the three instances described in Table 1 with both methods. For SEQ, we use the proposed crew pairing and crew assignment models by Zeighami and Soumis (2017). SEQ approach first solves the crew pairing problem and then crew assignment problem for a given set of pilots and copilots independently using CG. 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 is solved by CPLEX 12.4.

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

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

In Table 3, we compare the total CPU time (in minutes) needed to solve each of the three instances with the two approaches. For SEQ, the total CPU time can be divided into three parts: the time spent solving the

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

crew pairing problem, the time spent solving the pilot and copilot crew assignment problems sequentially. For INT the total CPU time includes the time to compute the initial partition, that is, the time to solve the crew pairing problem of Zeighami and Soumis (2017) and the overall time to solve ALR-pilot and ALR-copilot in all iterations. We report the total the total solution cost that was obtained with the two approaches. The last two columns of Table 3, respectively, give ratios between CPU times of INT and SEQ and the cost savings (in percentages) generated by the solutions produced by INT. These results show that INT can yield significant cost savings. In fact, for these instances, the total cost savings vary between 5.64%, 7.28%, and 7.49%, with an average of 6.80%. The good performance of the INT approach is that very few iterations suffice to find a good solution, the total iterations vary between 3 and 4. The main disadvantage of INT is the computational times that it yields, on average, 1.75 times longer than those obtained by SEQ.

Table 3: Computational time, cost for integrated and sequential approaches

	SEQ approach		INT approach				
Instance	CPU (min)	Cost	CPU (min)	Cost	Number of iterations	CPU INT/SEQ	Cost saving by INT (%)
1	5.47	1480089	11.42	1401069	3	2.08	5.64
2	19.56	1781845	36.28	1660930	4	1.85	7.28
3	61.50	2200857	80.15	2047500	4	1.30	7.48
Average						1.75	6.80

5.2 Computational gap in integrated and sequential approaches

In this experiment, we study the optimality gap of each instance that was obtained with the two approaches. The gap is the percentage difference between the LP and integer solutions of the problem. To find a lower bound, we stop CG at each branching node if the optimal value of the RMP improves by less than 0.1% over five iterations. The lower bound will improve during the branch and bound process since we explore many nodes. To obtain integer solutions, we use two branching heuristics as we explained in Section 4.4. The results of this experiment are given in Table 4. For SEQ, three optimality gaps are reported: the crew pairing gap corresponds to that for the crew pairing problem, the pilot and copilot gap correspond to that for the crew pilot and copilot assignment problems. For INT, two integrality gaps are reported, the first one corresponds to the solving ALR-pilot and the second one corresponds to the solving ALR-copilot. The results show that the percentage gaps are very small. The relatively small gaps indicate that the two approaches produce good solutions.

Table 4: Optimality gap (%) for integrated and sequential approaches

	SEQ approach			INT approach	
Instance	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
Average	0.93	0.21	0.14	0.02	0.01

5.3 Solution process of integrated approach

In this experiment, the goal is to study the solution process of INT in each iteration. Table 5 reports the percentage of similarity of pairings between two problems, the percentage of satisfied VRs and PFs, CPU time and the gap obtained in each iteration for each instance. One can observe that the good performance of INT is in part explained by the fact that for all three instances the percentage of similarity of pairings between two problems increases after each iteration while the percentage of satisfied VRs and PFs decrease very small. CPU time refer to the total effort needed to solve each instance in each iteration. For all instances, most of the CPU time is spent on the first iteration, and it can be explained as follows. Choosing a good initial solution has a very positive influence on the performance of DCA, it can result in less number of column generation iterations, hence, decrease the CPU time. We warm start the pilot and copilot problems after the first iteration. After solving each problem at each iteration we use the current optimal pairings as the initial solution for the next iteration. In particular, computation times are reduced considerably after first iteration. The results showed that there is a very significant reduction in computation time obtained after first iteration. CPU times were typically divided by a factor of three on each instance. The gap corresponds to the relative difference between the value of the lower bound and upper bound that obtained by ALR. The results show that the percentage gaps are small. The gap is between 0.00% and 0.01%, justifies the use of INT approach. For example, the first instance, the similarity of pairings in the first iteration is 92.44% and at the end of iteration three is 100%. The percentage of satisfied VRs for pilots and copilots is not decreased and the percentage of satisfied PFs is decreased very small, 0.66% and 0.60% for pilots and copilots respectively. The percentage gap in the first iteration is 1.45% while in the last iteration is 0.

Table 5: Solution process of INT

			Pilots		Copilots			
Instance	Iteration	Similarity pairings (%)	Satisfied VRs (%)	Satisfied PFs (%)	Satisfied VRs (%)	Satisfied PFs (%)	CPU (min)	Gap (%)
1	1	92.44	96.96	37.66	98.48	38.66	3.55	1.45
	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
2	1	89.50	92.64	43.65	97.05	41.50	12.15	1.75
	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
3	1	91.35	84.00	50.75	81.95	50.90	26.40	1.52
	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

5.4 Coverage of VRs and PFs

In this experiment, we study the performance of INT compare to SEQ in terms of the percentage of satisfied VRs and PFs. The results of this experiment are given in Tables 6 and 7. The results clearly show that INT has significantly better performance than SEQ. On average, INT increases the coverage of preferred vacation by 48.33% for pilots and 41.66% for copilots compared to SEQ. On average, INT increases the coverage of preferred flights by 15.33% for pilots and 12.66% for copilots compared to SEQ. These results are because the SEQ does not take into account the crew assignment constraints and objective during the building of the pairings, hence, the pairings generated by the crew pairing problem may not be suitable for the objective of the crew assignment problem. So, it is difficult to maximize the satisfaction of crew preference, VRs and PFs.

5.5 Similarity of pairings between two problems

In this experiment, we study the similarity of pairings between two problems for each approach. To reduce the propagation of the perturbations, pilots and copilots must have similar pairings as much as possible. Using SEQ to solve these problems produces the same pairings for pilots and copilots. Adding linking constraints in the integrated model helps us to reduce dissimilarity pairings between two problems while we use DCA. On average, INT decreases the similarity of pairings between two problems by -0.12% when compared with SEQ.

Table 6: Comparisons of the 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
Average					48.33	41.66

Table 7: Comparisons of the 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
Average					15.33	12.66

Table 8: Comparisons of the similarity pairings between two problems (%)

Instance	SEQ approach	INT approach	Deterioration by INT
1	100.00	100.00	0.00
2	100.00	99.65	-0.35
3	100.00	100.00	0.00
Average			-0.12

6 Conclusion

This paper has introduced an integrated crew pairing and personalized crew assignment model to generate personalized monthly schedules for a given set of pilots and copilots simultaneously. The proposed model keep the pairings in the two problems as similar as possible to reduce the propagation of the perturbations. We have presented an integrated approach based on combined ALR, CG, and DCA. This novel approach use Lagrangian decomposition in an alternating fashion proposing a new way to update Lagrangian multipliers. We studied real-world instances from a major US carrier. The integrated approach produced important savings with respect to a traditional sequential approach, crew pairing followed by crew assignment. Also, the results show that the integrated approach yields significant improvements compared with the sequential approaches in terms of satisfied VRs and PFs.

An interesting line of research to follow will be to generalization of the proposed alternating Lagrangian decomposition.

References

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

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.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. Column generation, vol. 5 of gerad 25th anniversary series, 2005.

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.

A. E. Eltoukhy, F. T. Chan, and S. Chung. Airline schedule planning: a review and future directions. Industrial Management & Data Systems, 117(6):1201–1243, 2017.

S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In Column generation, pages 33–65. Springer, 2005.

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, pages 1–27, 2014.

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 scheduling: A bi-dynamic constraint aggregation method using neighborhoods. European Journal of Operational Research, 212(3):445–454, 2011.

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.

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

P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. Operations Research, 45(2):188–200, 1997.

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.