CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Rolling Horizon Heuristic for Aircraft and Passenger Recovery

Karine Sinclair
Jean-François Cordeau
Gilbert Laporte

February 2016

CIRRELT-2016-05

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Rolling Horizon Heuristic for Aircraft and Passenger

Karine Sinclair[1,2,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
[2] Department of Logistics and Operations Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
[3] Department of Management Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. In the commercial airline industry, disruptions such as aircraft breakdowns, cancelled or delayed flights and airport capacity reductions can have a significant impact on the airline's planned operations and expenses. When disruptions occur, airlines must create new schedules so as to minimize the recovery cost and revert to the planned schedule as quickly as possible. To solve the joint aircraft and passenger recovery problem, we present a rolling horizon framework incorporating a column generation heuristic and show that the framework performs very well in solving the 2009 ROADEF Challenge instances, obtaining 12 of the best known solutions within 10 minutes of computing time. We also introduce new very large-scale instances and show that the rolling horizon framework is very effective in solving these instances within a reasonable computing time.

Keywords: Airline recovery, fleet assignment, aircraft routing, passenger itineraries, large neighbourhood search.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 2014-04959 and 2015-06189. This support is gratefully acknowledged.

---

* Corresponding author: Karine.Sinclair@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1. Introduction

The commercial airline industry is characterized by increased revenues and low profit margins. Between 1995 and 2014, system passenger revenues in the United States have almost doubled, reaching 126 billion dollars in 2014, while the average annual net income between 2000 and 2014 was −2.47 billion [1]. To optimize the use of the necessary ressources, operations research models and techniques have become widely used for schedule planning [6]. However, disruptions such as flight delays, flight cancellations, aircraft breakdowns, crew connection problems and unfavorable weather conditions can have a significant impact on the airline's planned operations and expenses. This is in great part due to the use of hub-and-spoke networks by most airlines, which causes small disruptions to have major repercussions on the network. When such disruptions occur, the airlines need to modify their planned schedule so as to minimize the total cost of recovery and return to the planned schedule by the end of the recovery period. Moreover, the recovery problem must be solved very quickly, usually within a few minutes, which increases the difficulty of solving large realistic instances. Because of the size and complexity of the problems considered, recovery problems are usually solved in a sequential manner. The first subproblem is the aircraft recovery problem which aims at creating new aircraft routes by cancelling or delaying flights, as well as modifying aircraft rotations. The crew recovery problem is then solved by reassigning crews, using either deadheading or reserve crews. Finally, the passenger recovery problem is solved by creating new passenger itineraries.

Teodorovic et al. [34] used a branch-and-bound heuristic to obtain optimal solutions to the homogenous fleet aircraft recovey problem, but no realistic instances were solved. The same problem was also solved by Jarrah et al. [18] and Arguello et al. [5]. The former developped minimum-cost network models which they solved by shortest path algorithms, while the latter developped a greedy randomized adaptive search procedure (GRASP) which selects candidate solutions and examines neighbouring solutions before proceeding with a local search phase. Eggenberg et al. [14] used a dynamic programming algorithm to generate the network model and applied a column generation heuristic to solve the heterogenous fleet aircraft recovery problem. Cao et al. [9, 10] modeled the problem as a quadratic programming problem and applied the approximate linear programming algorithm proposed by Hulburt [12], while Rosenberg et al. [29] modeled it as a set packing problem and reduced the size of the integer program by applying an aircraft selection heuristic. Dozic et al. [13] developped a three-phase heuristic, which removes flight legs on disrupted aircraft rotations, attempts to reinserts them in other rotations and interchanges parts of rotations between different aircraft. Finally, Xiuli et al. [36] developped a hybrid heuristic combining a tabu search heuristic and a GRASP.

Solving the aircraft recovery problem leads to additionnal disruptions in the crew scheduling and therefore the crew recovery problem is solved next. Stojković et al. [33] decomposed the problem as a set partitionning master problem, which they solved using a column generation method and a shortest path subproblem. Lettowsky et al. [20] modeled the problem as a set covering problem. The linear programming relaxation of the problem is solved first, and integer solutions are then found by branch-and-bound. Medard et al. [23] also modeled the problem as a set covering problem but used a greedy enumeration with a depth-first tree search approach and a column generation method. Other methods have been developped to solve the crew recovery problem; see, e.g., Abedlghany et al. [2], Nissen et al. [25] and Yu et al. [37].

Once the aircraft and crew recovery problems have been solved, the passenger recovery problem is usually addressed. To solve this problem, Bratu et al. [8] developped two models to determine the delayed flight legs and the cancelled flight legs, and applied a procedure to eliminate the dominated flight arc copies. Zhang et al. [40] developped a non-linear integer program that allows alternate transportation modes and developped an integrality constraint relaxation algorithm.

The decomposition of the recovery problem into subproblems can lead to suboptimal solutions. Therefore, recent research has focussed on integrating at least two of the three recovery problems. For the joint crew and aircraft recovery problem, Luo et al. [21] developped two models and used linear relaxation and a heuristic based on a restricted version of the model. Stojković et al. [33] presented the problem as a linear programming model and showed that the dual problem is a network flow problem that can be solved in linear time. Finally, Abdelghany et al. [3] developped a rolling horizon modeling framework that integrates a simulation model and a mixed integer programming model. The joint aircraft and passenger recovery problem has also been addressed. Zergodi et al. [38] developped a model for the aircraft recovery problem that considers the passenger delay and cancellation costs in the objective function and applied an ant colony algorithm, while Jafari et al. [16, 17] modeled the problem as a mixed integer program using aircraft rotations and passenger itineraries instead of flight legs. Finally, for the aircraft, crew and passenger recovery problem, Peterson et al. [28] applied a Benders decomposition scheme with the flight scheduling problem as the master problem and the recovery problems as the subproblems.

This paper presents a rolling horizon framework for the joint aircraft and passenger recovery problem as defined in the context of the 2009 ROADEF Challenge (Palpant et al. [26]). This problem consists of creating new aircraft routes and passenger itineraries so as to minimize the total cost and return to the planned schedule by the end of the recovery period. The different methodologies developped during the Challenge can be found on the website http://chalenge.roadef.org/2009. The following four methods produced the best solution for at least one instance. Bisaillon et al. [7], who ranked first in the Challenge, developped a large neighbourhood search heuristic that alternates between a construction, a repair and an improvement phase. Mansi et al. [22] proposed a two-phase algorithm that initially attempts to find a feasible solution using mixed integer programming and a repair heuristic, and then used an oscillation strategy alternating between a construction and a destruction phase. Peekstock et al. [27] developped a simulated annealing algorithm that handles aircraft, airport and passenger infeasibilities by introducing a penalty term in the objective function. Finally, Jozefowiez et al. [19] proposed a three-phase algorithm. The first phase integrates the schedule disruptions by removing flight legs and cancelling itineraries, while the second phase attempts to accommodate disrupted itineraries by assigning them to existing flight legs, and the third phase attempts to add new flight legs to the aircraft rotations.

After the Challenge, Acuna-Agost [4] developed a post-optimization procedure combined with the algorithm proposed by Jozefowiez et al. [19]. It uses network pruning algorithms to reduce the number of variables and constraints in the minimum cost multi-commodity flow model and proves capable of significantly improving solutions within a reasonable computing time. Sinclair et al. [31] proposed a large neighbourhood search heuristic (LNS) based on the Bisaillon et al. [7] algorithm, which, among other things, destroys and creates aircraft rotations so as to diversify the search. The proposed heuristic significantly improves the solution cost for 21 of the 22 ROADEF instances. Sinclair et al. [32] also developed a post-optimization algorithm which, when combined with the LNS heuristic, improves the solution cost for all instances of the ROADEF Challenge; it solves a mixed integer programming model and applies a column generation heuristic.

Whereas rolling horizon schemes have been applied to several fields of operations management (see Chand et al. [11] for an extensive review), their application to the passenger transportation literature is limited. Rolling horizon planning considers only a portion of the complete planning horizon at each step. After solving the problem for a given portion, only part of the decisions are fixed, while the others are optimized at the next step. This method is higly appropriate when the available information is limited, as in passenger rail transportation, where the origin-destination pairs of the passengers are unknown. Nielsen et al. [24] addressed a disruption management problem in the railway rolling stock problem and solved a mutli-commodity flow problem using a rolling horizon framework which reduces the computation time to a few seconds, even for large instances. Tornquist [35] applied a similar approach to the railway traffic disturbance problem.

When solving the airline recovery problem, all the necessary information is available at the beginning of the planning horizon. However, the use of a rolling horizon framework can still be appropriate in this context because of the complexity and the size of realistic instances, along with the limited computing time available. Zhang et al. [39] modeled the aircraft, crew and passenger recovery problem as a set partionning problem and solved it by means of a rolling horizon based algorithm, but the size of the instances considered was limited. As mentionned previously, Abdelghany et al. [3] developped a rolling horizon framework to solve the joint aircraft and crew recovery problem. For the joint aircraft and passenger recovery problem, Jafari et al. [16] presented an assignment model that considers flight delays, aircraft swapping, ferrying, the use of reserve aircraft and passenger cancellation or reassignment. The model was embedded within a rolling horizon framework but the size of the instances considered was limited. Saddoune et al. [30] developped a rolling horizon approach based on column generation to solve the crew pairing problem. They showed that the rolling horizon framework outperforms the three-phase approach traditionally applied in the industry. Furini et al. [15] proposed a rolling horizon approach that integrates a tabu search heuristic for the aircraft sequencing problem. Two mixed integer models were presented and rules for partitionning the aircraft sequence were proposed.

The first contribution of this article is to develop a rolling horizon scheme that allows the solution of large instances within a reasonable computing time. There exists a rich literature on airline recovery problems, but most research has focused on either small instances with realistic disruptions, or on large instances with very few disruptions. The second contribution is the creation of new very large realistic instances that take into consideration major disruptions on airport arrival and departure capacity.

The aircraft and passenger recovery problem is described in Section 2, followed by the solution methodology in Section 3. Section 4 presents the computational results, followed by conclusions and paths for future research in Section 5.

2. Problem description

The aircraft and passenger recovery problem aims at creating alternative aircraft routes and passenger itineraries when the airline schedules become infeasible due to disruptions. Airlines must construct these new aircraft routes and passenger itineraries in a timely manner so as to return to the original schedule by the end of the recovery period, while satisfying numerous constraints. The problem parameters are now described.

2.1. Airports

The airports are represented by the set of nodes $N = \{1, ..., n\}$, where each node corresponds to an airport at a given time. For each 60-minute time period begining at hour p, each airport i has given arrival and departure capacities defined as a_{ip} and b_{ip}, respectively.

2.2. Aircraft

Each aircraft f is characterized by an identification number, a family, a model and a cabin configuration. The fleet of aircraft operated by the airline is denoted by F. Aircraft belonging to the same family, for example AirbusSmall or BoeingSmall, have interchangeable crews. For each aircraft family there is a number of different models, for example models B747 and B777 belong to the BoeingBig family, and all aircraft of the same model have the same turn-round time, transit time and range. The cabin configuration is considered fixed and defines the capacity, denoted as cap_{mf}, of each cabin category m (economy, business and first) of aircraft f, with M being the set of all cabin classes. All aircraft of the same family, model and cabin configuration belong to the same group and let $\beta(f)$ be the index of the group to which aircraft f belongs. The aircraft can also have a planned maintenace, where they will become unavailable for a predetermined period of time. In such a case, the aircraft group $\beta(f)$ contains only one aircraft and is also characterized by the maintenance airport, the maintenance period and the number of flight hours before the maintenance, denoted by t_f. The demand for aircraft group $\beta(f)$ at node i is denoted by $d_{i\beta(f)}$.

2.3. Flight legs

Each aircraft f is assigned a rotation consisting of a sequence of flight legs that must satisfy the rotation continuity constraints and the turn-round time constraints. Flight legs are represented by the set of arcs $A = \{(i, j); i, j \in N, i \neq j\}$, where each arc (i, j) is a flight leg between nodes i and j or a connection arc between two nodes at the same airport. Arc (i, j) can also represent a ground transportation arc between two different airports within the same urban region. The set $A_{\beta(f)} = \{(i, j); i, j \in N, i \neq j\}$, is the set of arcs (i, j) associated with aircraft group $\beta(f)$. Flight legs are characterized by a travel time t_{ij} from i to j, and by the type of flight (domestic, continental or intercontinental). The set of arcs (i, j) arriving at node j during period p is defined as $O_{j(p)}$ while the set of arcs (i, j) departing from node i during period p is defined as $I_{i(p)}$. The set of aircraft F is partitioned into families, models and configurations. We define B_l as the set of all aircraft of family l in the set L of aircraft families, T_g as the set of all aircraft model g in the set G of aircraft models and V_h as the set of all aircraft configuration h in the set H of aircraft configurations.

2.4. Passenger reservations

The set of passenger reservations is denoted by $S = \{1, ..., s\}$, where each passenger reservation is characterized by the reservation number, the type of itinerary (inbound or outbound), the average price, the number of passengers and the itinerary, which consists of the sequence of flight legs that must satisfy the continuity constraints, the minimum connection time constraints and the cabin class constraint for each flight leg. The passenger demand for itinerary k is defined as d_k. For each itinerary k, we denote by D_k the set of all arcs (i, j) where i is the departure node of itinerary k, by U_k the set of all arcs (i, j) where i is the arrival node of itinerary k and by W_k the set of all dummy arcs (i, j), which represent the cancellation arcs between the departure and the arrival nodes for itinerary k. Passenger itineraries must also satisfy additionnal functionnal constraints. The first are the delay constraints, which ensure that the maximum delay does not exceed 18 hours for domestic and continental itineraries and 36 hours for international itineraries. These constraints do not apply to return itineraries. The second are the origin-destination constraints which ensure that passengers on a reconstructed itinerary depart from the original departure airport and arrive at the original destination airport. Finally, passengers on a reconstructed itinerary cannot depart before the original departure time.

2.5. Disruptions

Four types of disruptions are considered: flight delays, flight cancellations, airport disruptions and aircraft disruptions. Airport disruptions limit the number of departures and arrivals at a given airport for a given time period, whereas aircraft disruptions lead to the unavailability of an aircraft for a period of time. In the latter case, no flight legs can be assigned to the aircraft during the disruption period.

2.6. Problem formulation

The problem can be represented on a time-space network $G = (N, A)$, where N is the set of nodes and A is the set of arcs previously defined. The objective is to minimize the total recovery cost, which consists of delay costs, cancellation costs, downgrading costs, flight operating costs and costs for non-compliant location of aircraft. There are two types of delay costs, direct costs and passenger disutility costs. The former consist of the disbursements the airline incurs in case of delays (lodging, drinks and meal expenses) and vary depending on the extent of the delay and the initial planned duration. The passenger disutility costs measure the passenger's perceived disturbance. It depends on the extent of the delay, the reservation type and the itinerary's reference cabin class. The delay costs are denoted by c^{del}_{ijmk}. The cancellation cost c^{can}_k of itinerary k is the reimbursement of the ticket price as well as a financial compensation that varies depending on the initially planned duration of the itinerary. Downgrading costs c^{down}_{km} are also a measure of the disutility of passengers and are incurred when the cabin class of one or more flight legs of the new itinerary is lower than the itinerary's reference cabin class. These costs depend on the type of flight leg and on the level of downgrading. The operating costs c^{op}_{ijf} depend on the travel time t_{ij} from node i to node j. They are considered to be independent of the number of passengers and of the possible delays. Finally, costs for non-compliant location of aircraft are incurred when the required number of aircraft is not available at the specified airports by the end of the recovery period.

There are three types of non-compliance penalties. A penalty cost c^{fam} is incurred whenever a required aircraft cannot be matched with an aircraft of the same family, whereas a penalty cost c^{mod} is incurred if a required aircraft cannot be matched with an aircraft of the same model. Finally, a penalty cost c^{conf} is incurred if a cabin configuration cannot be matched. Parameters n_j^l, n_j^g and n_j^h are the number of aircraft of family l, the number of aircraft of model g and the number of aircraft of configuration h requested at node j at the end of the recovery period. D^{fam}, D^{mod} and D^{conf} are duplicates of the final node at each airport and represent, respectively, the family demand node, the model demand node and the configuration demand node, and the sets R_l, R_g and R_h are, respectively, the sets of all arcs between these nodes and a dummy node.

3. Solution methodology and mathematical model

In order to solve the joint aircraft and passenger recovery problem we have developped a rolling horizon variant of the column generation post-optimization heuristic described by Sinclair et al. [32]. Using the column generation post-optimization heuristic proves time consuming for large instances. Therefore, we have embedded the column generation heuristic within a rolling horizon framework, thus reducing the computing time and memory needed. We will first summarize the column generation heuristic developped by Sinclair et al. [32], and then describe the rolling horizon framework.

3.1. Column generation post-optimization heuristic

The column generation post-optimization heuristic is initialized with the solution obtained by applying the LNS heuristic developped by Sinclair et al. [31]. This solution is included in the initial restricted master problem of the column generation heuristic.

3.1.1. Large neighbourhood search heuristic

The LNS heuristic of Sinclair et al. [31] is an improvement to the LNS heuristic developped by Bisaillon et al. [7] for the ROADEF 2009 Challenge. The heuristic alternates between three phases: construction, repair and improvement.

Construction phase

In the first phase, the heuristic attempts to obtain an initial feasible solution by delaying or cancelling flights. If a rotation becomes infeasible because of delays on previous flights in the rotation, the first flight yielding an infeasibility is delayed by increments of 60 minutes. If no feasible rotation can be found, the flight is declared critical and will be treated in the repair phase. This phase also considers rotation infeasibility due to a cancelled flight in the rotation. The heuristic attempts to recreate the cancelled flight. If this proves impossible, the smallest sequence of flights that forms a loop, starting with the flight that yields the infeasibility, is removed. If no such loop exists, the flight yielding the infeasibility and all the subsequent flights in the rotation are cancelled. Rotation infeasibility due to maintenance is treated by removing a loop before the maintenance that allows the rotation to respect the maintenance constraints. Flights that are cancelled because of airport capacity are rescheduled at a later, less congested period. If this is not possible, either the smallest loop is removed or all subsequent flights in the rotation are removed. Finally, when a rotation becomes infeasible due to aircraft breakdowns, the heuristic tries to recreate all the disrupted flights by means of a longest path algorithm.

Repair phase
The first step of the repair phase considers the critical flights obtained in the construction phase because of delays on previous flights, and tries to delay them to a less congested period. If this proves impossible, either the smallest loop containing the flight or the sequence starting from this flight to the end of the rotation is removed. Once aircraft rotation feasibility has been regained, the heuristic attempts to reinsert flight sequences that have been previously removed in the rotations of available aircraft. In the last two steps, the heuristic attempts to reaccommodate passengers whose itineraries have been cancelled, first by repeatedly solving a shortest path problem, then by creating new flights by either solving a multi-commodity flow problem or a shortest path problem.

Improvement phase
First, the improvement phase performs a local search to improve the solution by delaying some flights so as to accommodate additional passengers. The passengers are reassigned by repeatedly solving the shortest path problem. The heuristic then diversifies the search by destroying either part of a rotation or a complete rotation, and by then creating new flights using all of the available aircraft, again by repeatedly solving a shortest path problem. The next step attempts to reacommodate passengers at disrupted airports (airports with reduced arrival and departure capacity). Passengers are assigned to an existing flight departing from the disrupted airports and the heuristic attempts to create new flights from the flight's arrival airport to

the passengers's destination airport. In the opposite way, the heuristic also tries to accommodate passengers whose final destination is a disrupted airport. Because of the maximum delay constraints, it is sometimes impossible to accommodate passengers with a cancelled itinerary on a certain flight leg. The heuristic attempts to delay subsequent passengers who have the same origin-destination pair, thus accommodating additional passengers. Finally, new flights are created by means of a shortest path algorithm so as to redirect the aircraft to the required airports at the end of the recovery period.

3.1.2. Column generation heuristic

Sinclair et al. [32] developped the following mixed integer progamming model for the aircraft and passenger recovery problem (APRP), where x_{ijf} = 1 if and only if arc (i,j) is operated by aircraft f, and y_{ijmk} is the number of passengers from itinerary k assigned to arc (i,j) in class m.

(APRP)

$$\text{minimize} \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{km}^{down} y_{ijmk} \tag{1}$$

$$+ \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{ijmk}^{del} y_{ijmk} \tag{2}$$

$$+ \sum_{(i,j)\in W_k} \sum_{m\in M} \sum_{k\in K} c_{k}^{can} y_{ijmk} \tag{3}$$

$$+ \sum_{(i,j)\in A} \sum_{f\in F} c_{ijf}^{op} x_{ijf} \tag{4}$$

$$+ \sum_{f\in B^l} \sum_{(i,j)\in R^l} c^{fam} x_{ijf} \tag{5}$$

$$+ \sum_{f\in T^g} \sum_{(i,j)\in R^g} c^{mod} x_{ijf} \tag{6}$$

$$+ \sum_{f\in V^h} \sum_{(i,j)\in R^h} c^{conf} x_{ijf} \tag{7}$$

subject to

$$\sum_{i\in N,(i,j)\in A_{\beta(f)}} x_{ijf} - \sum_{i\in N,(j,i)\in A_{\beta(f)}} x_{jif} = 0 \qquad \beta(f) \in F,\ j \in N \text{ and } d_{j\beta(f)} = 0 \tag{8}$$

$$\sum_{j \in N, (i,j) \in A_{\beta(f)}} x_{ijf} \leq 1 \qquad \beta(f) \in F,\ i \in N \text{ and } d_{i\beta(f)} = 1 \tag{9}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^l} x_{jif} \leq n_j^l \qquad f \in F,\ j \in N,\ l \in L \text{ and } j = D^{fam} \tag{10}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^g} x_{jif} \leq n_j^g \qquad f \in F,\ j \in N,\ g \in G \text{ and } j = D^{mod} \tag{11}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^h} x_{jif} \leq n_j^h \qquad f \in F,\ j \in N,\ h \in H \text{ and } j = D^{conf} \tag{12}$$

$$\sum_{(i,j) \in I_j(p)} \sum_{f \in \beta(f)} \sum_{\beta(f) \in F} x_{ijf} \leq a_{jp} \qquad p \in P,\ j \in N \tag{13}$$

$$\sum_{(i,j) \in O_i(p)} \sum_{f \in \beta(f)} \sum_{\beta(f) \in F} x_{ijf} \leq b_{ip} \qquad p \in P,\ i \in N \tag{14}$$

$$\sum_{f \in F} x_{ijf} \leq 1 \qquad (i,j) \in A \tag{15}$$

$$\sum_{(i,j) \notin D_k U_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k U_k} \sum_{m \in M} y_{jimk} = 0 \qquad k \in K, j \in N \tag{16}$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \qquad k \in K \tag{17}$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{mf} x_{ijf} \qquad (i,j) \in A,\ m \in M \tag{18}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ijf} \leq t_f \qquad f \in F \tag{19}$$

$$y_{ijmk} \geq 0 \qquad (i,j) \in A,\ m \in M,\ k \in K \tag{20}$$

$$x_{ijf} = \ 0 \text{ or } 1 \qquad (i,j) \in A,\ f \in F. \tag{21}$$

The terms (1)–(3) of the objective function define, respectively, the downgrading costs, the delay costs and the cancellation costs, while term (4) defines the aircraft operating costs. The terms (5)–(7) are, respectively, the non-compliance costs for the incorrect family, model and configuration positionning. Constraints (8) and (9) ensure that the aircraft flow conservation constraints are respected, while constraints (10)–(12) are the aircraft flow conservation constraints for the family, the model and the configuration, respectively. Constraints (13) and (14) enforce the airport capacity limit. Constraints (15) are the assignment constraints which ensure that each flight arc is assigned to at most one aircraft. Constraints (16) and (17) enforce passenger flow conservation. Finally, Constraints (18) ensure that the aircraft seating capacity limits are respected, and constraints (19) ensure that each flight does not exceed the maximum flight hours.

The initial restricted master problem (RMP) includes all aircraft and passenger arcs found in the LNS solution. To solve the problem by column generation, the linear programming (LP) relaxation of the problem is considered and the dual variables π^{ik}, λ^k and $\mu^{ijm} \leq 0$ are associated with constraints (16)–(18). At each iteration, a percentage of variables with negative reduced cost are included in the LP relaxation and this phase is repeated a predetermined number of iterations. Then the APRP problem is solved.

In order to solve the larger instances, the size of the model is reduced by considering all passengers with the same origin-destination pair, the same departure time and the same cabin class as a passenger group. Also, all aircraft with the same characteristics are aggregated except for aircraft with planned maintenance. Finally, for some instances, only a subset of itineraries are considered. Since solving the APRP problem for the large instances is impractical, Sinclair et al. [32] use the column generation heuristic just described to solve the following multi-commodity flow problem for passengers (MC-APRP), where $X^* = \{(i,j,f)\colon\ (i,j) \in A, f \in F \text{ and } x_{ijf} = 1\}$ is the set of the aircraft arcs found in the LNS solution.

(MC-APRP)

$$\text{minimize} \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{km}^{down} y_{ijmk} \tag{22}$$

$$+ \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{ijmk}^{del} y_{ijmk} \tag{23}$$

$$+ \sum_{(i,j)\in W_k} \sum_{m\in M} \sum_{k\in K} c_{k}^{can} y_{ijmk} \tag{24}$$

subject to

$$x_{ijf} = 1 \qquad (i,j,f) \in X^* \tag{25}$$

$$\sum_{(i,j)\notin D_k U_k} \sum_{m\in M} y_{ijmk} - \sum_{(j,i)\notin D_k U_k} \sum_{m\in M} y_{jimk} = 0 \qquad k \in K, j \in N \tag{26}$$

$$\sum_{(i,j)\in D_k} \sum_{m\in M} y_{ijmk} = d_k \qquad k \in K \tag{27}$$

$$\sum_{k\in K} y_{ijmk} \leq \sum_{f\in F} cap_{fm} x_{ijf} \qquad (i,j) \in A,\ m \in M \tag{28}$$

$$y_{ijmk} \geq 0 \qquad (i,j) \in A,\ m \in M,\ k \in K \tag{29}$$

$$x_{ijf} = \ 0 \text{ or } 1 \qquad (i,j) \in A,\ f \in F. \tag{30}$$

The terms (22)–(24) are, respectively the downgrading costs, the delay costs and the cancellation costs. Constraints (25) are the assignment constraints, which ensure that all aircraft arcs found in the LNS solution take value 1. Constraints (26) and (27) are the passenger flow conservation constraints, and constraints (28) enforce the aircraft seeting capacity limit.

3.2. Rolling horizon

The column generation heuristic described in the previous section is embedded in a rolling horizon framework depicted in Figure 1. At iteration t, the recovery period is divided into rolling periods Δ_t which are in turn divided in a fixed and non-fixed part ($\Delta_t^{'}$ and $\Delta_t^{''}$, respectively). At the first iteration, all itineraries and all flights departing within the rolling period Δ_1 are considered. Since many passenger itineraries contain more than one flight, limiting the flights considered to those departing before the end of the rolling horizon period would limit the number of possibilities for the itineraries with departures close to the end of the rolling horizon period. Therefore, in order to accommodate additionnal passengers, flights departing within an additionnal time period Θ are also considered.

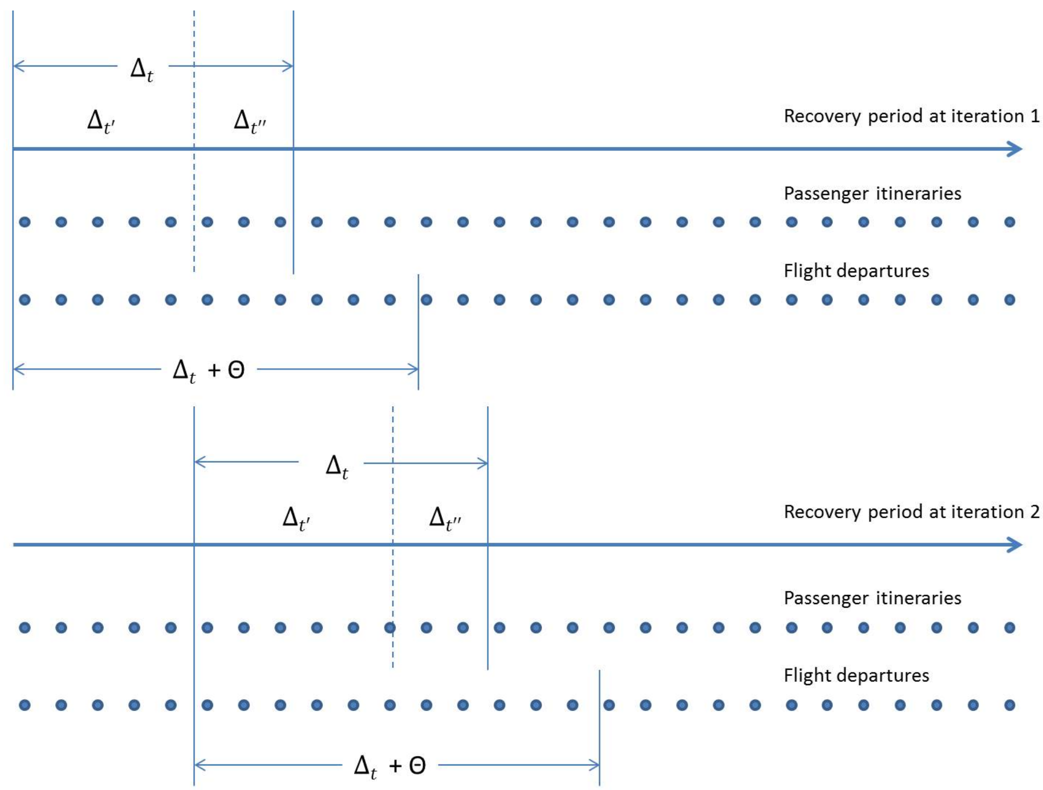

Figure 1: Rolling horizon framework

The column generation heuristic is applied in each rolling horizon period and the LP relaxation of the APRP problem is solved for a fixed number of iterations, denoted by α, before solving the APRP problem at the final iteration $\alpha+1$. At each iteration, a percentage of the variables with negative reduced costs is included in the LP relaxation. The itineraries obtained in the column generation heuristic's solution that begin within the fixed part of the rolling period Δ_t', are fixed. At iteration, $t+1$, all itineraries from the previous rolling period that remain cancelled are considered in the following rolling period Δ_{t+1}, which includes the non-fixed part of the previous rolling period Δ_t, and the column generation heuristic is applied again. The rolling horizon framework ends when the end of the rolling period is equal to or greater than the end of the recovery period.

4. Computational results

The rolling horizon heuristic was implemented and tested on a computer with two Intel Westmere EP X5650 six-core processors running at 2.667 GHz and with 96 GB of memory. Our algorithm was tested on the instances of the 2009 ROADEF Challenge, described in Tables 1–5, as well as on some new, very large instances derived from the 2009 ROADEF Challenge instances.

4.1. The 2009 ROADEF Challenge Instances

To test our methodology we used the instances provided in the 2009 ROADEF Challenge. Tables 1 and 2 present the disruption characteristics of the B instances. For these instances, the recovery period is 36 hours. There are 256 aircraft, 45 airports, 1,423 flights and 11,214 passenger itineraries.

Table 1 Characteristics the B01–B05 instances

	B01	B02	B03	B04	B05
Flight disruptions	230	255	229	230	0
Aircraft disruptions	0	0	1	0	0
Airport disruptions	0	0	0	1	34

Table 2 Characteristics the B06–B10 instances

	B06	B07	B08	B09	B10
Flight disruptions	230	255	229	230	77
Aircraft disruptions	0	0	1	0	0
Airport disruptions	0	0	0	1	34

Tables 3–5 present the characteristics of the XA, XB and X instances. The smaller instances, the XA instances, contain 85 aircraft, 35 airports and 608 flights. The XB instances have 256 aircraft, 44 or 45 airports and 1,423 flights, while the larger instances, the X instances, contain 618 aircraft, 168 airports and 2,178 flights.

Table 3 Characteristics of the XA instances

	XA01	XA02	XA03	XA04
Recovery period (h)	14	52	14	52
Itineraries	1,943	3,959	1,872	3,773
Flight disruptions	83	0	83	
Aircraft disruptions	3	3	3	3
Airport disruptions	0	407	0	407

Table 4 Characteristics of the XB instances

	XB01	XB02	XB03	XB04
Recovery period (h)	36	52	36	52
Itineraries	11,214	11,214	11,565	11,565
Flight disruptions	229	0	228	0
Aircraft disruptions	3	1	4	4
Airport disruptions	0	34	0	34

Table 5 Characteristics of the X instances

	X01	X02	X03	X04
Recovery period (h)	78	78	78	78
Itineraries	28,308	28,308	29,151	29,151
Flight disruptions	0	0	0	0
Aircraft disruptions	1	1	1	1
Airport disruptions	1	0	1	0

4.2. Creation of very large instances derived from the 2009 ROADEF Challenge instances

As can be seen in Tables 1–5, four instances from the Challenge are very large (i.e. instances X01, X02, X03 and X04), with a recovery period of 78 hours, 618 aircraft, 168 airports, 2,178 flights and a number of passenger itineraries varying between 28,308 and 29,151. Although the size of these instances is realistic, they contain very few disruptions (no flight disruptions, one aircraft disruption and either no or one airport disruption). We have therefore created new instances with numerous disruptions, based on these large instances. Since instance X02 is identical to X01 and X04 is identical to X03 except for the disruptions, only instances X01 and X03 were used to create the new instances.

We reduced the recovery period to 55 hours so as to be able to create flight disruptions during the first day of the schedule. The disruption scenarios created are similar to those in the Challenge in terms of flight delays, flight cancellations, aircraft disruptions and airport disruptions. We used four scenarios of flight disruptions (no disruption, 111 disruptions, 161 disruptions and 215 disruptions, noted -F, 111F, 161F and 215F respectively), four scenarios of airport disruptions (100% airport capacity, 75% airport capacity, 50% airport capacity and 30% airport capacity, noted -P, 75P, 50P and 30P respectively) and three scenarios of aircraft disruptions (one disruption, two disruptions and four disruptions, noted 1C, 2C and 4C respectively).

We created 48 instances from instance X01 and 48 instances from instance X03 using all combinations of the disruption scenarios described above. All instances can be found at http://chairelogistique.hec.ca/data/airline/ and are identified as follows: X0175P2C111F is the instance X01 with the following disruption combination: 75% airport capacity, two aircraft disruptions and 111 flight disruptions. A description of all instances is presented in Appendix 1.

4.3. Parameter setting

To calibrate the different parameters we used a total of 24 instances: the 12 instances derived from the X01 instance with 30% airport arrival and departure capacity, refered to as the X0130P instances, and the 12 instances derived from the X03 instance with 75% airport arrival and departure capacity, referred to as the X0375P instances.

4.3.1 Number of iterations and percentage of variables with negative reduced cost

For the parameter setting, the number of iterations performed within the column generation heuristic and the percentage of variables included in the LP relaxation are considered simultaneously. The number of iterations considered varies between three and six, and the percentage of variables varies between 40% and 100%. Table 6 presents the average improvement with respect to the best solution as well as the average computing time for all possible combinations of these parameters for the X0130P instances and the X0375P instances. The solutions obtained with 4 iterations and 100% of the variables with negative reduced cost included in the LP relaxation offer a good compromise between solution quality and computing time.

Table 6 Average solution improvement and computing times

	3 iterations		4 iterations		5 iterations		6 iterations	
	% improvement	time	% improvement	time	% improvement	time	% improvement	time
40%	16.84%	1965.63	16,72%	2135.50	4.251%	2298.71	2.06%	2481.58
60%	9.09%	1913.24	2.37%	2195.42	1.04%	2380.92	0.83%	2547.25
80%	6.94%	2028.79	1.31%	2262.67	0.54%	2436.50	0.623%	2598.04
100%	3.194%	2090.00	0.86%	2299.75	0.50%	2481.33	0.594%	2654.04

4.3.2 The fixed rolling horizon period

The rolling horizon period Δ defines the period for which passenger itineraries are considered in the column generation heuristic at each iteration. All itineraries departing before the end of the rolling horizon period are considered, but only part of the solution's itineraries are fixed. The rolling horizon period can therefore be considered as a fixed part, Δ', and a non-fixed part, Δ''. The average solution gap and the average computing time for the different percentages of the rolling horizon that are fixed are reported in Figures 2 and 3, respectively. The number of iterations within the column generation heuristic was set to four and the percentage of variables with negative reduced cost included in the LP relaxation was set at 100%. The best average computing time and the best average gap with respect to the best solution was obtained when parameter Δ' was set to 80% of Δ.

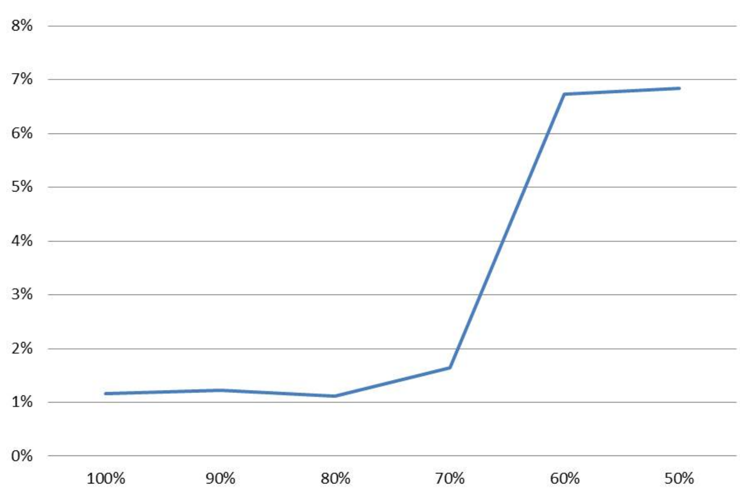

Figure 2: Improvement with respect to best solution value for parameter Δ'

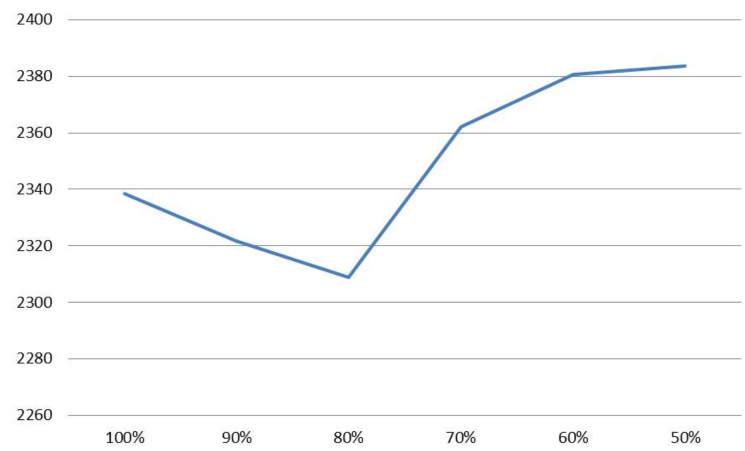

Figure 3: Computing time in seconds for parameter $\Delta^{'}$

4.3.3 The additional time period for flights Θ

The additionnal time period considered for the flights is defined as Θ. Figure 4 presents the average gap with respect to the best solution when the value of Θ varies between 0 and 1200 minutes, while Figure 5 presents the average computing time for the same values of Θ. The number of iterations within the column generation heuristic was set to four, the percentage of variables with negative reduced costs included in the LP relaxation was set to 100% and $\Delta^{'}$ was set at 80% of Δ. As can be seen in Figure 4, solutions improve when we increase Θ from 0 minutes to 500 minutes. Then, solution quality slowly deteriorates. Figure 5 shows that the computing time decreases when Θ is increased. The value of Θ was therefore set to 500 minutes.

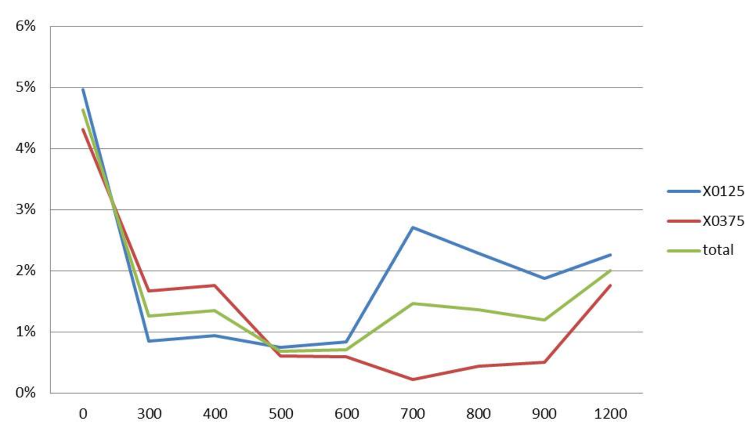

Figure 4: Improvement with respect to best solution value for parameter Θ

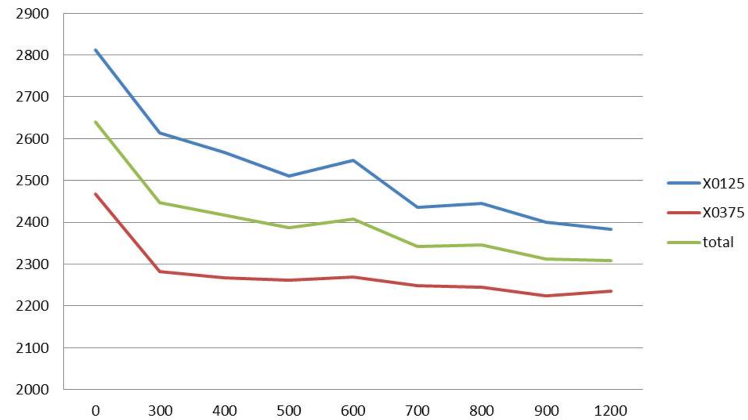

Figure 5: Computing time in seconds for parameter Θ

4.3.4 Length of the rolling horizon

The length of the rolling horizon period Δ has an impact on solution quality as well as on computing time. Tables 7–9 present the average gap with respect to the LNS solution value obtained after 10 minutes of computing time and 40 minutes of computing time for the X0130P instances and the X0375P instances as well as the average computing time for rolling horizon lengths varying between 900 and 1,500 minutes. Again, the number of iterations within the column generation heuristic was set to four, the percentage of variables with negative reduced costs included in the LP relaxation was set to 100% and Δ' was set at 80% of Δ. Finally, the additional time period for flights Θ was set to 500 minutes. The best solution values are obtained with a rolling horizon period of 1,200 minutes.

Table 7 Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	900 minutes		
	% improvement wrt LNS 40 min	% improvement wrt LNS 40 min	Seconds
X0125P	9.72%	-2.04%	2378.00
X0375P	9.49%	2.43%	2013.92

Table 8 Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	1,200 minutes		
	% improvement wrt LNS 40 min	% improvement wrt LNS 40 min	Seconds
X0125P	13.66%	2.43%	2508.75
X0375P	15.66%	9.94%	2271.58

Table 9 Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	1,500 minutes		
	% improvement wrt LNS 40 min	% improvement wrt LNS 40 min	Seconds
X0125P	11.51%	-0.01%	2637.33
X0375P	14.98%	9.59%	2393.25

4.4. Results

4.4.1 New instances

Since the rolling horizon heuristic is a post-optimization heuristic, we have run the LNS heuristic for 10 minutes before executing the rolling horizon heuristic. In order to compare the quality of the solutions obtained with the rolling horizon heuristic for the 98 new instances, we ran the LNS heuristic for the average computing time of the rolling horizon heuristic, which includes the 10 minutes for the LNS heuristic. To this end, we have separated the instances into eight groups: the X01 instances with 100% airport capacity, the X01 instances with 75% airport capacity, the X01 instances with 50% airport capacity and the X01 instances with 30% airport capacity, the X03 instances with 100% airport capacity, the X03 instances with 75% airport capacity, the X03 instances with 50% airport capacity and the X03 instances with 30% airport capacity (denoted by X01-P, X0175P, X0150P, X0130P, X03-P, X0375P, X0350P and X0330P respectively). The rolling horizon length Δ was set to 1200 minutes, Δ' was set to 80% of Δ, Θ was set to 500 minutes, the number of iterations within the column generation heuristic was set to four and 100% of the variables with negative reduced costs were included in the LP relaxation.

For each group, we ran the LNS heuristic for the average computing time of the rolling horizon heuristic for that group. Table 8 presents the average gap with respect to the solution values obtained by means of the LNS heuristic when ran for 10 minutes and when ran for the average computing time of the rolling horizon heuristic for the respective group. The solution costs for all instances are reported in Appendix 2. As can be seen from Table 8, when there are no airport disruptions, as for the X01-P instances and the X03-P instances, the LNS heuristic performs well, outperforming the rolling horizon heuristic for the X01-P instances. However, the rolling horizon heuristic performs very well for all instances with airport capacity disruptions.

Table 10 Average solution improvement wrt LNS 10 minutes and LNS average time

	Average time seconds	% improvement wrt LNS 10 min	% improvement wrt LNS average time
X01-P	1913	1.21%	-0.49%
X0130P	2508	13.66%	1.57%
X0150P	2453	13.64%	9.77%
X0175P	2038	8.00%	4.10%
X03-P	1950	1.49%	0.63%
X0330P	3049	7.21%	4.63%
X0350P	2739	13.05%	5.94%
X0375P	2271	15.66%	18.02%

4.4.2 The 2009 ROADEF Challenge instances

Additionnal parameter setting tests were performed for the rolling horizon length parameter for the 2009 ROADEF Challenge instances. Tables 11 and 12 present the solution values and computing time for the rolling horizon heuristic, as well as solution values when running the LNS heuristic for 10 minutes, and the gap with respect to the LNS solution value.

Table 13 presents the best known solution values for the 2009 ROADEF Challenge instances obtained within 10 minutes of computing time as well as the rolling horizon heuristic solution value including the LNS heuristic obtained within 10 minutes of computing time (total time of 10 minutes for the LNS heuristic and rolling horizon heuristic). The rolling horizon length used is 1800 minutes. The heuristic was able to find solutions within 10 minutes for 14 of the 22 instances and obtained 12 best solution values. It is important to note that the speed of the computer used is faster than the one used during the 2009 ROADEF Challenge. Reducing the computing time to take into consideration the speed of the computer would not yield good feasible solutions.

Table 14 presents the solution values obtained with the column generation heuristic of Sinclair et al. [32] as well as the computing time and the improvement with respect to the solution values obtained when running the LNS for 10 minutes. It is important to note that the LNS heuristic was run for 10 minutes before the column generation heuristic was applied. Table 14 also presents the solution values obtained with the rolling horizon heuristic and the computing time. The improvements with respect to the LNS solution values and the improvements with respect to the column generation heuristic are also reported in Table 14. Solutions found with the column generation outperform the rolling horizon heuristic, which was expected since the column generation heuristic is identical to the rolling horizon heuristic with a horizon length equal to the total recovery time. However the computing time necessary to obtain good solutions is much faster for the rolling horizon heuristic, except for instances X01–X04, which require a longer computing time but yield better solution costs.

5. Conclusions and future research

We have developed a rolling horizon framework for the post-optimization column generation heuristic to solve the joint aircraft and passenger recovery problem. The rolling horizon framework has proven very effective on the instances of the 2009 ROADEF Challenge with 10 minutes of computing time, yielding 12 out of 22 best solutions. We have also presented new, very large instances with many disruptions and showed that the rolling horizon framework performs very well on the instances with important airport disruptions, and outperforms the LSN heuristic of Sinclair et al. [31].

Future research should focus on developping useful tools for the airline industry that incorporate the rolling horizon framework by allowing an initial solution to be applied to the aircraft rotations and to the passenger itineraries with the earliest departure times, while solving the passenger recovery problem for the remainder of the passenger itineraries. It should eventually integrate the crew recovery problem to yield large realistic recovery problems.

Acknowledgments This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 2014-04959 and 2015-06189. This support is gratefully acknowledged.

Table 11 Rolling horizon heuristic solution costs

		900 min			1,200 min			1,500 min		
	LNS 10 min	Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS
B01	870812	699172	305	19.71%	732792	294	15.85%	712561	309	18.17%
B02	1056280	921709	274	12.74%	880891	294	16.60%	929974	299	11.96%
B03	931289	846095	286	9.15%	738978	314	20.65%	776523	362	16.62%
B04	962451	893814	287	7.13%	768812	282	20.12%	817409	310	15.07%
B05	8377616	5288122	617	36.88%	5402406	702	35.51%	4675439	747	44.19%
B06	2829589	2402166	309	26.11%	2179034	313	22.99%	2141244	340	24.33%
B07	3157541	2603379	317	17.55%	2449561	312	22.42%	2495971	367	20.95%
B08	3134454	2601498	300	17.00%	2394693	321	23.60%	2378236	365	24.13%
B09	2676733	2464949	318	7.91%	2048043	354	23.49%	2012754	376	24.81%
B10	32984637	29861715	831	9.47%	26795800	914	18.76%	22707143	938	31.16%
X01	142443	79753	2883	44.01%	INF.			INF.		
X02	-20964	-45867	2834	118.79%	INF.			INF.		
X03	1353315	1032698	3041	23.69%	INF.			INF.		
X04	145534	82853	2949	43.07%	INF.			INF.		
XA01	79711	72280	51	9.32%	72280	51	9.32%	72280	51	9.32%
XA02	1664971	1483525	106	10.90%	1488596	111	10.59%	1141325	118	31.45%
XA03	262945	231755	72	11.86%	231755	73	11.86%	231755	73	11.86%
XA04	4440451	4299784	124	3.17%	4190474	129	5.63%	3349620	138	24.57%
XB01	1220781	1086093	281	11.03%	1060394	289	13.14%	1111985	319	8.91%
XB02	8319761	5203970	618	37.45%	5144276	702	38.17%	4531979	729	45.53%
XB03	4659228	3957083	319	15.07%	3490899	348	25.08%	3585616	379	23.04%
XB04	33759687	31136330	845	7.77%	28714742	872	14.94%	24106526	944	28.59%

Table 12 Rolling horizon heuristic solution costs

		1,800 min			2,100 min		
	LNS 10 min	Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS
B01	870812	705949	338	18.93%	652546	333	25.06%
B02	1056280	849771	330	19.55%	849715	351	19.56%
B03	931289	714124	390	23.32%	714150	397	23.32%
B04	962451	734609	322	23.67%	734514	330	23.68%
B05	8377616	4490626	750	46.40%	4588308	792	45.23%
B06	2829589	1909322	388	32.52%	1909677	384	32.51%
B07	3157541	2164230	428	31.46%	2168023	432	31.34%
B08	3134454	2085396	382	33.47%	2085346	407	33.47%
B09	2676733	1827216	427	31.74%	1830889	437	31.60%
B10	32984637	23243194	1019	29.53%	25619963	919	22.33%
X01	142443	INF.			INF.		
X02	-20964	INF.			INF.		
X03	1353315	INF.			INF.		
X04	145534	INF.			INF.		
XA01	79711	72280	51	9.32%	72280	51	9.32%
XA02	1664971	1229003	123	26.18%	1123739	132	32.51%
XA03	262945	231755	73	11.86%	231755	73	11.86%
XA04	4440451	3292092	146	25.86%	3351022	146	24.53%
XB01	1220781	981685	340	19.95%	981685	635	19.59%
XB02	8319761	4466412	753	46.32%	4489754	784	46.04%
XB03	4659228	2977349	387	36.10%	2977566	433	36.09%
XB04	33759687	24793086	1052	26.56%	27245307	913	19.30%

Table 13 Improvement wrt the best known solution

	Best know solution value	Cost	% improvement wrt Best
B01	843983	710673	15.80%
B02	1220708	904519	25.90%
B03	859809	753967	12.31%
B04	968796	793647	18.08%
B05	8717434	INF.	
B06	2733781	1998689	26.89%
B07	4588857	2371871	48.31%
B08	3067543	2084658	32.04%
B09	2678741	1894053	29.29%
B10	32171455	INF.	
X01	222040	INF.	
X02	-18917	INF.	
X03	1316835	INF.	
X04	240656	INF.	
XA01	64799	72280	-11.54%
XA02	1462311	1148697	21.45%
XA03	186279	231755	-24.41%%
XA04	3972419	3560533	10.37%
XB01	1028223	981449	4.55%
XB02	9986653	INF.	
XB03	4204317	3276451	22.07%
XB04	33326556	INF.	

Table 14 Rolling horizon heuristic, column generation heuristic and LNS heuristic solution costs

		Column generation			Rolling horizon			
	LNS 10 min	Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS	Computing time improvement wrt CG
B01	870812	505382	788	41.96%	652546	333	25.06%	57.74%
B02	1056280	679362	782	35.68%	849715	351	19.56%	55.12%
B03	931289	549928	824	40.95%	714124	390	23.32%	52.66%
B04	962451	604307	790	37.20%	734514	330	23.68%	58.23
B05	8377616	4357412	1351	47.99%	4490626	750	46.40%	44.49%
B06	2829589	1608877	1168	43.14%	1909322	388	32.52%	66.78%
B07	3157541	2848062	1417	9.80%	2164230	428	31.46%	69.90%
B08	3134454	1668229	1260	46.78%	2085396	382	33.47%	69.98%
B09	2676733	1512454	1202	43.50%	1827216	427	31.74%	64.48%
B10	32984637	22304388	2062	32.37%	22707143	938	31.16%	54.51%
X01	142443	125019	1456	12.23%	79753	2883	44.01%	-98.01%
X02	-20964	-36857	1179	75.81%	-45867	2834	118.79%	-140.04%
X03	1353315	1217999	1299	10.00%	1032698	3041	23.69%	-134.10%
X04	145534	140356	1288	3.56%	82853	2949	43.07%	-128.96%
XA01	79711	64799	108	18.71%	72280	51	9.32%	52.78%
XA02	1664971	951385	209	42.86%	1123739	132	32.51%	36.84%
XA03	262945	186279	91	29.16%	231755	73	11.86%	19.78%
XA04	4440451	2697195	302	39.26%	3292092	146	25.86%	51.66%
XB01	1220781	647134	855	46.99%	981685	340	19.95%	60.23%
XB02	8319761	4641007	1317	44.22%	4466412	753	46.32%	42.82%
XB03	4659228	2469046	1346	47.01%	2977349	387	36.10%	71.25%
XB04	33759687	22973144	1892	31.96%	24106526	944	28.95%	50.11%

Appendix 1

Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions	Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions
X01-P1C-P	100%	1	0	X0150P1C-P	50%	1	0
X01-P1C111P	100%	1	111	X0150P1C111P	50%	1	111
X01-P1C161P	100%	1	161	X0150P1C161P	50%	1	161
X01-P1C215P	100%	1	215	X0150P1C215P	50%	1	215
X01-P2C-P	100%	2	0	X0150P2C-P	50%	2	0
X01-P2C111P	100%	2	111	X0150P2C111P	50%	2	111
X01-P2C161P	100%	2	161	X0150P2C161P	50%	2	161
X01-P2C215P	100%	2	215	X0150P2C215P	50%	2	215
X01-P4C-P	100%	4	0	X0150P4C-P	50%	4	0
X01-P4C111P	100%	4	111	X0150P4C111P	50%	4	111
X01-P4C161P	100%	4	161	X0150P4C161P	50%	4	161
X01-P4C215P	100%	4	215	X0150P4C215P	50%	4	215
X0175P1C-P	75%	1	0	X0130P1C-P	30%	1	0
X0175P1C111P	75%	1	111	X0130P1C111P	30%	1	111
X0175P1C161P	75%	1	161	X0130P1C161P	30%	1	161
X0175P1C215P	75%	1	215	X0130P1C215P	30%	1	215
X0175P2C-P	75%	2	0	X0130P2C-P	30%	2	0
X0175P2C111P	75%	2	111	X0130P2C111P	30%	2	111
X0175P2C161P	75%	2	161	X0130P2C161P	30%	2	161
X0175P2C215P	75%	2	215	X0130P2C215P	30%	2	215
X0175P4C-P	75%	4	0	X0130P4C-P	30%	4	0
X0175P4C111P	75%	4	111	X0130P4C111P	30%	4	111
X0175P4C161P	75%	4	161	X0130P4C161P	30%	4	161
X0175P4C215P	75%	4	215	X0130P4C215P	30%	4	215

Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions	Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions
X03-P1C-P	100%	1	0	X0350P1C-P	50%	1	0
X03-P1C111P	100%	1	111	X0350P1C111P	50%	1	111
X03-P1C161P	100%	1	161	X0350P1C161P	50%	1	161
X03-P1C215P	100%	1	215	X0350P1C215P	50%	1	215
X03-P2C-P	100%	2	0	X350P2C-P	50%	2	0
X03-P2C111P	100%	2	111	X0350P2C111P	50%	2	111
X03-P2C161P	100%	2	161	X0350P2C161P	50%	2	161
X03-P2C215P	100%	2	215	X0350P2C215P	50%	2	215
X03-P4C-P	100%	4	0	X0350P4C-P	50%	4	0
X03-P4C111P	100%	4	111	X0350P4C111P	50%	4	111
X03-P4C161P	100%	4	161	X0350P4C161P	50%	4	161
X03-P4C215P	100%	4	215	X0350P4C215P	50%	4	215
X0375P1C-P	75%	1	0	X0330P1C-P	30%	1	0
X0375P1C111P	75%	1	111	X0330P1C111P	30%	1	111
X0375P1C161P	75%	1	161	X0330P1C161P	30%	1	161
X0375P1C215P	75%	1	215	X0330P1C215P	30%	1	215
X0375P2C-P	75%	2	0	X0330P2C-P	30%	2	0
X0375P2C111P	75%	2	111	X0330P2C111P	30%	2	111
X0375P2C161P	75%	2	161	X0330P2C161P	30%	2	161
X0375P2C215P	75%	2	215	X0330P2C215P	30%	2	215
X0375P4C-P	75%	4	0	X0330P4C-P	30%	4	0
X0375P4C111P	75%	4	111	X0330P4C111P	30%	4	111
X0375P4C161P	75%	4	161	X0330P4C161P	30%	4	161
X0375P4C215P	75%	4	215	X0330P4C215P	30%	4	215

Appendix 2

Table 17 Rolling horizon heuristic solution costs for X01-P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X01-P1C-F	29202	29202	28877	1911	1,11%	1,13%
X01-P1C111F	563630	563630	563305	1899	0,06%	0,06%
X01-P1C161F	835229	835229	834904	1913	0.04%	0.04%
X01-P1C215F	933434	933434	933109	1913	0.03%	0.03%
X01-P2C-F	165120	165120	162645	1905	1.50%	1.50%
X01-P2C111F	699548	699548	697073	1904	0.35%	0.35%
X01-P2C161F	971147	971147	968672	1907	0.25%	0.25%
X01-P2C215F	1069353	1069353	1066878	1895	0.23%	0.23%
X01-P4C-F	912728	841019	812728	1919	10.96%	3.48%
X01-P4C111F	1449564	1378187	1449564	1923	0.00%	-4.92%
X01-P4C161F	1721160	1649784	1721160	1946	0.00%	-4.15%
X01-P4C215F	1819369	1747992	1819369	1923	0.00%	-3.92%
				1913	1.21%	-0.49%

Table 18 Rolling horizon heuristic solution costs for X0175P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0175P1C-F	3153374	3004508	2757677	2043	12.55%	8.95%
X0175P1C111F	3572524	3371889	3197068	2034	10.51%	5.47%
X0175P1C161F	3844123	3643488	3468915	2052	9.76%	5.03%
X0175P1C215F	3946908	3745664	3568266	2074	9.59%	4.97%
X0175P2C-F	3272176	3134469	2895885	2007	11.50%	8.24%
X0175P2C111F	3739109	3529792	3362360	2048	10.08%	4.98%
X0175P2C161F	4010708	3801250	3636468	2046	9.33%	4.53%
X0175P2C215F	4109453	3900437	3734436	2031	9.13%	4.45%
X0175P4C-F	4279559	4105679	4054053	2052	5.27%	1.27%
X0175P4C111F	4718534	5497474	4576832	1996	3.00%	0.45%
X0175P4C161F	5214131	5104734	5074184	2026	2.68%	0.60%
X0175P4C215F	5321330	5199999	5184522	2050	2.57%	0.30%
				2038	8.00%	4.10%

Table 19 Rolling horizon heuristic solution costs for X0150P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0150P1C-F	21948616	21363891	19406537	2503	11.58%	10.09%
X0150P1C111F	20671318	20878103	18151722	2546	12.19%	15.02%
X0150P1C161F	20564631	21414615	17513827	2525	14.84%	22.27%
X0150P1C215F	20666937	21519807	17626017	2432	14.71%	22.09%
X0150P2C-F	21992638	17078081	18842310	2391	14.32%	-9.36%
X0150P2C111F	20486160	19604211	16889389	2511	17.56%	16.07%
X0150P2C161F	21356782	21292165	18076663	2342	15.36%	17.79%
X0150P2C215F	20431422	21457855	17254349	2388	15.55%	24.36%
X0150P4C-F	21157806	18535435	18528213	2417	12.43%	0.04%
X0150P4C111F	22190477	19932935	18256991	2455	13.22%	3.51%
X0150P4C161F	22718027	20173250	20392381	2455	10.24%	-1.07%
X0150P4C215F	23171850	19736721	20459564	2475	11.71%	-3.53%
				2453	13.64%	9.77%

Table 20 Rolling horizon heuristic solution costs for X0125P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0175P1C-F	34675423	29410233	23230736	2498	15.70%	0.61%
X0175P1C111F	36074841	32182535	31414598	2517	12.92%	2.44%
X0175P1C161F	36513843	32464402	31895637	2508	12.65%	1.78%
X0175P1C215F	36298541	32825097	31719284	2507	12.62%	3.49%
X0175P2C-F	34408636	30235969	29519861	2431	14.21%	2.43%
X0175P2C111F	35976686	31986739	31019605	2591	13.78%	3.12%
X0175P2C161F	35944436	32020919	31260576	2542	13.03%	2.43%
X0175P2C215F	36680910	32435885	32295562	2531	11.96%	0.43%
X0175P4C-F	36507512	31032889	31326811	2491	14.19%	-0.94%
X0175P4C111F	36887223	32138180	31195609	2478	15.43%	3.02%
X0175P4C161F	37200872	32345717	31966348	2507	14.07%	1.19%
X0175P4C215F	37815066	32393931	32757815	2504	13.37%	-1.11%
				2508	13.66%	1.57%

Table 21 Rolling horizon heuristic solution costs for X03-P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X03-P1C-F	135214	130872	135214	1959	0.00%	-3.21%
X03-P1C111F	752125	474783	752125	1946	0.00%	-0.58%
X03-P1C161F	1073334	1068992	1073334	1941	0.00%	-0.40%
X03-P1C215F	1166308	1161966	1166308	1967	0.00%	-0.37%
X03-P2C-F	352510	340002	318013	1977	9.79%	6.91%
X03-P2C111F	974273	961909	941039	1971	3.41%	2.22%
X03-P2C161F	1295475	1283015	1261931	1966	2.59%	1.67%
X03-P2C215F	1386865	1374399	1357188	2019	2.14%	1.27%
X03-P4C-F	3516158	3516158	3516158	1937	0.00%	0.00%
X03-P4C111F	4133069	41133069	4133069	1906	0.00%	0.00%
X03-P4C161F	4454272	4454272	4454272	1909	0.00%	0.00%
X03-P4C215F	4547252	4547252	4547252	1909	0.00%	0.00%
				1951	1.49%	0.63%

Table 22 Rolling horizon heuristic solution costs for X0375P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0375P1C-F	8545854	9018845	7446975	2119	12.86%	21.11%
X0375P1C111F	9945090	8892441	7687516	2287	20.70%	15.67%
X0375P1C161F	10452293	9265976	8212186	2476	21.43%	12.83%
X0375P1C215F	10556691	9149021	8305761	2441	21.32%	10.15%
X0375P2C-F	8860841	10087113	7519444	2190	15.14%	34.15%
X0375P2C111F	9453074	10626522	7960659	2167	15.79%	33.49%
X0375P2C161F	10700057	9513342	8587105	2298	19.75%	10.79%
X0375P2C215F	10793242	9237398	8680608	2270	19.57%	6.41%
X0375P4C-F	14118152	13749637	12393596	2263	12.22%	10.94%
X0375P4C111F	12996004	14073368	11193765	2298	13.98%	25.74%
X0375P4C161F	12240808	14582552	11133030	2219	9.05%	30.98%
X0375P4C215F	14447469	14390462	13837715	2231	4.22%	3.99%
				2271	15.66%	18.02%

Table 23 Rolling horizon heuristic solution costs for X0350P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0350P1C-F	53059175	47022208	45155550	2709	14.90%	4.13%
X0350P1C111F	51079460	47594942	44666606	2790	12.55%	6.56%
X0350P1C161F	50778584	47584786	44955426	2853	11.47%	5.85%
X0350P1C215F	51650387	46312204	45294260	2828	12.11%	2.02%
X0350P2C-F	51451902	52048527	42151673	2649	18.08%	23.48%
X0350P2C111F	52597896	47124646	45416808	2712	13.65%	3.76%
X0350P2C161F	53489259	47168467	46274118	2722	13.49%	1.93%
X0350P2C215F	53779458	47541232	46359710	2657	13.80%	2.55%
X0350P4C-F	51442930	53119278	45013115	2780	12.50%	18.01%
X0350P4C111F	56579786	51511186	50461435	2719	11.34%	2.69%
X0350P4C161F	57376533	50069568	51606809	2714	10.06%	-2.89%
X0350P4C215F	58196178	52495569	50843306	2745	12.63%	3.25%
				2740	13.05%	5.94%

Table 24 Rolling horizon heuristic solution costs for X0325P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0325P1C-F	81339237	83705344	76190805	3074	6.33%	9.86%
X0325P1C111F	84435683	81597879	76959762	2969	8.85%	6.03%
X0325P1C161F	85665316	82611641	81329258	2987	5.06%	1.58%
X0325P1C215F	84758414	80621689	78722522	3257	7.12%	2.41%
X0325P2C-F	85079528	83260844	78743529	2965	7.45%	5.74%
X0325P2C111F	82112646	82148598	73756072	3026	10.18%	11.38%
X0325P2C161F	85409599	82935582	78539683	3081	8.04%	5.60%
X0325P2C215F	87438471	81817686	83111907	2959	4.95%	-1.56%
X0325P4C-F	89604072	85356624	82970773	3252	7.40%	2.88%
X0325P4C111F	89879808	84263755	84290957	2945	6.22%	-0.03%
X0325P4C161F	90616014	88196817	84545997	3014	6.70%	4.32%
X0325P4C215F	88925649	87678986	81632612	3060	8.20%	7.41%
				3049	7.21%	4.63%

References

[1] databank.worldbank.org/data/.

[2] A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127:309–331, 2004.

[3] K. F. Abdelghany, A. F. Abdelghany, and G. Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185:825–848, 2008.

[4] R. Acuna-Agost. Passenger improver (PI). A new approach for the passenger recovery problem. POSTDOC Report, AMADEUS and École nationale de l'aviation civile, Toulouse, 2010.

[5] M.F. Arguello, J.F. Bard, and G. Yu. A GRASP for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 5:211–228, 1997.

[6] C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37:368–391, 2003.

[7] S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin. A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR: A Quarterly Journal of Operations Research*, 9:139–157, 2011.

[8] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9:279–298, 2006.

[9] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part I: Mathematical formulation. *Transportation Planning and Technology*, 20:183–199, 1997.

[10] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part II: Algorithm and computational experiments. *Transportation Planning and Technology*, 20: 201–217, 1997.

[11] S. Chand, V. Hsu, and S. Sethi. Forecast, solution and rolling-horizons in operations management problems: a classified bibliography. *Manufacturing and Service Operations Management*, 4:25–43, 2002.

[12] T.F. Coleman and L.A. Hulburt. A direct active set algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming*, 45:373–406, 1989.

[13] S. Dozić, M. Kalić, and O. Babić. Heuristic approach to the airline schedule disturbances problem: single fleet case. *Procedia-Social and Behavioral Sciences*, 54:1232–1241, 2012.

[14] N. Eggenberg, M. Salani, and M. Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Computers & Operations Research*, 37:1014–1026, 2010.

[15] F. Furini, M. P. Kidd, C. A. Persiani, and P. Toth. Improved rolling horizon approaches to the aircraft sequencing problem. *Journal of Scheduling*, pages 1–13, 2015.

[16] N. Jafari and S. H. Zegordi. The airline perturbation problem: considering disrupted passengers. *Transportation Planning and Technology*, 33: 203–220, 2010.

[17] N. Jafari and S. H. Zegordi. Simultaneous recovery model for aircraft and passengers. *Journal of the Franklin Institue*, 348:1638–1655, 2011.

[18] A.I.Z. Jarrah and G. Yu. A decision support framework for airline cancellations and delays. *Transportation Science*, 27:266–280, 1993.

[19] N. Jozefowiez, C. Mancel, and F. Mora-Camino. A heuristic approach based on shortest path problems for integrated flight, aircraft and passenger rescheduling under disruptions. *Journal of the Operational Research Society*, 64:384–395, 2012.

[20] L. Lettovsky, E. L. Johnson, and G. L. Nemhauser. Airline crew recovery. *Transportation Science*, 34:337–348, 2000.

[21] S. Luo and G. Yu. On the airline schedule perturbation problem caused by the ground delay program. *Transportation Science*, 31:298–311, 1997.

[22] R. Mansi, S. Hanafi, C. Wilbault, and F. Clautiaux. Disruptions in the airline industry: math-heuristic for re-assiging aircraft and passengers simultaneously. *European Journal of Industrial Engineering*, 6:690–712, 2012.

[23] C. P. Medard and N. Sawhney. Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183:1013–1027, 2007.

[24] L.K. Nielsen, L. Kroon, and G. Maroti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220:465–509, 2012.

[25] R. Nissen and K. Haase. Duty-period-based network model for crew rescheduling in European airlines. *Journal of Scheduling*, 9:255–278, 2006.

[26] M. Palpant, M. Boudia, C-A. Robelin, S. Gabteni, and F. Laburthe. ROADEF 2009 Challenge: Disruption management for commercial aviation. Amadeus S.A.S, Operations Research Division, Sophia Antipolis, France, www.roadef.org. 2009.

[27] J. Peekstok and E. Kuipers. ROADEF 2009 Challenge: Use of a simulated annealing-based algorithm in disruption management for commercial aviation.
http://challenge.roadef.org/2009/files/PeekstokKuipers.pdf.

[28] J. D. Petersen, G. Solveling, E. L. Johnson, J.-P. Clarke, and S. Shebalov. An optimization approach to airline integrated recovery. *Transportation Science*, 46:482–500, 2012.

[29] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37:408–421, 2003.

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

[31] K. Sinclair, J.-F. Cordeau, and G. Laporte. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233:234–245, 2014.

[32] K. Sinclair, J.-F. Cordeau, and G. Laporte. A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. *Computers & Operations Research*, 65:42–52, 2015.

[33] M. Stojković, F. Soumis, J. Desrosiers, and M. M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A*, 36:779–788, 2002.

[34] D. Teodorović and S. Guberinić. Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15:178–182, 1984.

[35] J. Törnquist. Railway traffic disturbance management - an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research Part A*, 41:249–66, 2007.

[36] Z. Xiuli and G. Yanchi. An improved GRASP for irregular flight recovery. System Science and Engineering (ICSSE), 2012 International Conference on. 2012.

[37] G. Yu, M. Arguello, G. Song, S.M. McCowan, and A. White. A new era for crew recovery at Continental Airlines. *Interfaces*, 33:5–22, 2003.

[38] S. H. Zegordi and N. Jafari. Solving the airline recovery problem by using ant colony optimization. *International Journal of Industrial Engineering & Production Research*, 23:121–128, 2010.

[39] D. Zhang and Y.K. Lau. A rolling horizon based algorithm for solving the integrated airline schedule recovery problem. *Journal of Automation and Control Engineering*, 2:232–237, 2014.

[40] Y. Zhang and M. Hansen. Real-time inter-modal substitution (RTIMS) as an airport congestion management strategy. *Transportation Research Records*, 2052:90–99, 2008.