Les cahiers Leibniz

Disruption management for commercial aviation, a mixed integer programming approach

Julien Darlay, Louis-Philippe Kronek, Susann Schrenk, Lilia Zaourar

Laboratoire G-SCOP
46 av. Félix Viallet, 38000 GRENOBLE, France
ISSN : 1298-020X

n° 183

July 2010

Site internet : http://www.g-scop.inpg.fr/CahiersLeibniz/

Disruption Management for Commercial Aviation, A Mixed Integer Programming Approach

J. Darlay, Louis-Philippe Kronek, Susann Schrenk, Lilia Zaourar

July 2, 2010

Abstract

In this paper we describe our approach to the airline disruption problem which was the topic of the ROADEF'09 challenge. The aim of the challenge was to re-assign aircrafts and passengers simultaneously in case of disruption, w.r.t. maintenance constraints. We decompose the problem into two main steps: re-assignment of aircrafts on flights, and re-accommodation of passengers on operated flights. Both steps take profit of the initial optimized plan, such as passengers are already involved in aircraft re-assignment. These steps are simply modelled as multicommodity network flows with side constraints and solved using a MIP-solver. We report the numerical results on the instances provided during that challenge. These results have been quite improved since the end of the challenge with just a little modification in our program.

1 Introduction

Commercial airlines operate flights according to a published flight schedule that is optimized from a revenue standpoint. However, external events, such as mechanical failures, personnel strikes, or bad weather conditions, frequently occur, disrupting planned airline operations. In such case, it is necessary to find efficient solutions that minimize the duration of the disruption, thus minimizing its impact.

Traditionally, resources are re-allocated sequentially, according to a natural hierarchy: aircraft, crew, passengers. Nonetheless, this method has serious flaws. Namely, decision making at the local level, concerning one resource, can lead to global repercussions, affecting all resources. For example, a change in the flight schedule, potentially effecting aircraft resources, may also lead to a missed connection for crew or for passengers. Thus, an increasing effort is being made to integrate different levels of decision making. The topic proposed for this challenge follows this trend, since it aims at re-assigning aircrafts and passengers simultaneously in case of disruptions. This problem is still a simplified version of the real problem since nearly all constraints associated with crews and ground-staff are ignored.

The problem is more complex than the academic problem generally addressed in the literature. In fact, most papers on disruption management consider only one aspect of the problem. Generally, in the disruption management literature passengers are given a low priority [8]. By far most work on operational recovery problems has been reported on the aircraft resource. Since crews can be repositioned fairly easily and standby crews are often available, the aircraft are seen as the scarce resource. The aircraft recovery problem has been tackled through network flow models [6], multicommodity flow formulation with side constraints [1, 3], local search [10], metaheuristics [2], column generation [12] and different ad-hoc heuristics. Far less work is published on the crew recovery problem, which is far more complex due to the number of cabin crews and the more complex rules and regulations for crews. There is less work published on the passenger recovery problem [9]. Recovery of multiple resources was first considered in [13]. This solution considers aircrafts, crews and maintenance. To modify the fleet assignment due to the fluctuating passenger demand pattern, the possibility of modifying the daily airline schedule and assignment is discussed in [7]. This can be viewed as a fleet/passenger recovery model. They use a network flow model to generate a new fleet assignment without considering crews and maintenance. A recent contribution [4] describes two models for integrated recovery. The models are focused on passenger recovery but are also incorporating a certain extent rules and regulations on aircraft and crew, e.g. reserve crews are incorporated but not the ability to recover a disrupted crew. For a more in-depth and theoretical description of the academic research within disruption management we refer to the review [5].

Our aim is to present a heuristic approach for disruption management for commercial aviation. This work was initiated within the 2009 ROADEF Challenge which addressed the disruption management for commercial aviation encountered by the leading provider of IT solutions to tourism and travel industry, AMADEUS. Our team ranked fifth out of 29 registered teams, and got the second prize of the junior category (at the beginning of the challenge, none of us had his PhD). The problem can viewed as a fleet/passenger recovery problem including maintenance constraints. We develop a heuristic based on a decomposition of the problem into two main steps as follows: aircraft recovery with respect to maintenance, and re-assignment of passengers on operated flights. These steps are then applied sequentially, but both take profit of the initial optimized plan. Thus, passengers are already involved in the aircraft recovery. Both steps are simply modelled as a multicommodity network flow model with side constraints respectively with arc- (path-) based formulations. Moreover, this decomposition and models, easily allow the integration of new constraints in one of the steps, which permits to construct a feasible solution, without need of a correction procedure afterwards. Thus the main contribution of this paper is to tackle algorithmically the difficult distribution management for commercial aviation, with simple multicommodity network flow models with side constraints, solved using a MIP solver. Furthermore, the numerical results on the instances provided during that challenge show the competitiveness of our approach. With a simple modification on our program, we improve our solutions and obtain a final score close to the best of the final phase.

The paper is organized as follows. Section 2 introduces the problem and notations. Section 3 presents our algorithmic approach to solve the problem. It describes the multicommodity network flow problems with side constraints used for the construction of the new rotation plan and for re-accommodation of passengers. Section 4 reviews the computational results. Conclusions and directions for future research follow in Section 5.

2 Problem definition

In this section we briefly sketch the main characteristics of our problem. For more details on input data, costs, constraints and instances, the reader is referred to the full problem description [11].

2.1 Data

The goal of the challenge is to resume normal operations as quickly as possible, in case of disruptions to the planned flight schedule. The flight schedule that was originally planned has to be applied without any modification after a given period of time, called *recovery period*. Airports, aircrafts, and passengers are the key players of the problem. Flights and rotations link these three players. More precisely, the following information are given in any instance of the problem:

- *Airports and capacities*: Airports are characterized by given restrictions, such as the maximum departure and arrival rates. For each airport, the number of arrivals and departures are limited; the airport arrival and departure capacities are given as one-hour interval capacities.
  The *distance between airports* is the nominal flight time between each airport pair, the flight type (domestic, continental or intercontinental) depends upon these airports.

- *Aircrafts and maintenance*: The fleet is composed of the set of aircrafts operated by an airline. Each aircraft is defined by: an identification number, a model, and a cabin configuration. Operational characteristics, such as turn-round time, transit time, are given. They are common to all aircrafts of a given model (e.g. Boeing 747, A320, ...). Subsets of aircrafts with common characteristics are grouped within *families* (e.g., A318, A319, A320, and A321 are in the "AirbusSmall" family). We also have the description of required maintenance for concerned aircrafts.
  Aircraft positions at end of recovery: the number of aircrafts of each kind (model and configuration) required to be at any given airport by the end of the recovery period is given.
- *Itineraries*: The description of passenger itineraries includes a unique identification number, the nature of the itinerary (inbound or outbound), the unit price in euros, the number of passengers booked on this itinerary, the series of flights, the dates and the cabin classes (business (B), first (F) and economic (E)) constituting the itinerary.
- *Flights and rotations*: We have the full description of the rotations of all aircrafts throughout the recovery period. Each flight is uniquely defined by a flight number, origin and destination airports, departure and arrival days and times and by the aircraft operating the flight.

Following incidents from a perturbation on the rotation plan are considered:

- *Flight disruptions* are either delays or cancellations.
- Each *aircraft disruption* corresponds to the unavailability of an aircraft for a given period.
- *Airport disruptions* correspond to reduction of airport departure and arrival capacities.

A solution of the problem is a set of rotations and itineraries that describe respectively information for all entries of the original flights and rotations, including modifications and new itineraries for each passenger, and considering the disruptions. The updated flight schedule is the result of decisions regarding flights planned for the original schedule: intentional *cancellations* and *delays*, *aircraft changes* within a given family and the possible *additions* of flights.
The objective of the problem is to find a solution of minimum cost respecting all imperative constraints. This cost includes both, operational costs and costs modelling passengers' inconvenience. They are briefly defined in section 2.3 after the description of the constraints in the next section.

2.2 Constraints and violation penalties

2.2.1 Imperative constraints

Constraints are associated with aircrafts, airports, itineraries, and rotations. All these constraints are imperative, violation of one of these constraints leads to an unfeasible solution. They are described as follows:

- **Constraints on aircrafts**:
  - *Maintenance and unavailability*: maintenance constraint enforces the concerned aircraft to be at a given airport at the beginning of the required maintenance. Then, the aircraft has to stay in that airport for the maintenance duration. Moreover, the global flight time of the aircraft before its maintenance is limited. An *unavailable aircraft* has to stay on ground on its whole unavailability period.
  - *Aircraft seating capacities*: Of course, for each flight the number of passengers travelling in each cabin class must not exceed the seating capacity of that cabin.
  - *Aircraft changes*: A change of aircraft operating a flight can only be done within the aircrafts of the same family.

- *Minimum turn-round time*: two consecutive flights of the same aircraft rotation must be separated by at least a turn-round time corresponding to the aircraft operating the flights.

- **Constraints on airports**:
  - *Airport capacities*: The airport capacity constraints enforce an upper bound of the number of departures and arrivals for each one-hour interval at each airport.

- **Constraints on rotations**:
  - *Recovery period constraint*: Rotations that have arrived or have already left at the beginning of the recovery period in the initial flight schedule cannot be modified at all.

- **Constraints on itineraries**: These constraints concern passengers whose itineraries have been modified, if these constraints cannot all be respected, the passengers trip has to be cancelled.
  - *Minimum connection time*: the minimum connection time constraint must be respected for all connecting passengers. Two consecutive flights on the itinerary of a passenger must be separated by at least 30 minutes at the connection airport.
  - For each modified itinerary, we must have the same origin and final destination as in the original itinerary.
  - The modified itinerary must not start before the time of the first flight of the original itinerary.
  - *Maximum total delay* at destination must not exceed 18 hours for domestic and continental flights, and 36 hours for an intercontinental flight.

Any violation of one of these constraints leads to an unfeasible solution. In practice, the most difficult constraints to satisfy are maintenance constraints combined with airport departure and landing capacities.

2.2.2 Violation penalties

Aircraft positions at end of recovery are the only non imperative constraints. Any violation of these constraints induces a cost (penalty). Other constraints, as for example the priority in re-accommodation of passengers who have upcoming connecting flights and have already started their trip at a given time or passengers with an itinerary corresponding to the return portion of their trip is a consequence of the cost structure on passenger re-accommodation. A non respect of these constraints leads not to an unfeasible solution; it only affects the cost of the solution.

2.3 Cost function

The objective function includes parameters related to additional operating costs or gains due to the modification of the flight plan, penalties for non compliant aircraft location at the end of the recovery period, as well as a measure of the disutility to passengers. The objective is to minimize a weighted sum of those factors.

Operating costs include all costs associated with the operation of a flight. On one hand they are composed of 15 euros for drink and meal costs, 60 euros for lodging cost, depending on the length of the delay. In the case of cancellation, the airline must reimburse the ticket price regardless of the length of the trip, as well as provide financial compensation. On the other hand, they include a penalty in the case of non-compliant location of aircraft at the end of the recovery period. The value of the penalty depends upon the seriousness of the violations: violation of configuration, violation on model, violation on family.

The *disutility to a passenger* measures the disturbance perceived by the passenger, regardless of the compensation mentioned above. The disutility is expressed in euros, as a function of the total delay as compared to the original itinerary. The delay cost is linear in the total delay at destination, with a slope depending upon itinerary type (type of the longest flight leg) and the itinerary reference cabin class (highest of the booking cabin class on the different flight legs). These costs include also a penalty in case of downgrading (change to a lower cabin class on all or part of the trip).

2.4 Notations and problem parameters

Sets:
A = set of all airports indexed by i or j.
K = set of all aircrafts, indexed by k. KM (respectively KU) is the subset of aircrafts with a maintenance (respectively an unavailability)
F = set of all flights, indexed by f.
D = set of all possible additional delays, indexed by d.
H = set of all hours, indexed by h.
T = set of all times (discretized in minutes) starting at time 0, ending at time TE (End Recovery), indexed by t.
P = set of all itineraries of passengers (aggregated), indexed by p.
CL = set of different cabin classes business (B), first (F) and economic (E), indexed by cl.
Q^p = set of all possible itineraries for re-accommodation of passengers of itinerary p, indexed by q.

Data and parameters:
Δ_t = time discretization step (5 minutes).
τ^k = transit time of aircraft k.
c^k_{fd} = cost for using aircraft k on flight f delayed by delay d.
c_f = cost for cancelling flight f.
$cap^k(cl)$ = seat capacity in cabin class cl of aircraft k.

AD_f, AA_f, AD^p, AA^p, AD^k, AA^k = departure and arrival airports of flight f, passengers from itinerary p, and aircraft k respectively.
TD_f, TA_f, TD^p, TA^p = departure and arrival time of flight f, and passengers of itinerary p, respectively.

$CapTakeOff[i,h]$ = maximum number of departures from airport i in the one hour interval $[h, h+1[$.
$CapLanding[j,h]$ = maximum number of arrivals at airport j in the one hour interval $[h, h+1[$.

TMB^k, TME^k = beginning time and end time of maintenance of aircraft k.
AM^k = airport of maintenance of aircraft k.
$MaxFlightHours^k$ = maximum number of flight hours before the beginning of the required maintenance.
TUB^k, TUE^k = beginning and end time of unavailability of aircraft k.

n^p = number of passengers of itinerary p.
$MaxDelayPassenger^p$ = maximum allowed delay at arrival for passengers of itinerary p.
C^p_q = cost for re-accommodating one passenger of itinerary p on itinerary q.
CC^p = cost for cancelling the trip of one passenger of itinerary p.
δ^{kcl}_{fq} = 1 if flight f operated by aircraft k considering cabin class cl is in itinerary q.

Decision variables:
x^k_{fd} = 1 if flight f is operated by aircraft k with a delay d.
X_f = 1 if flight f is cancelled.
y^k_{it} = 1 if aircraft k stays in airport i between time t and $t + \Delta_t$.

z_q^p = number of passengers of p re-accommodated on itinerary q.

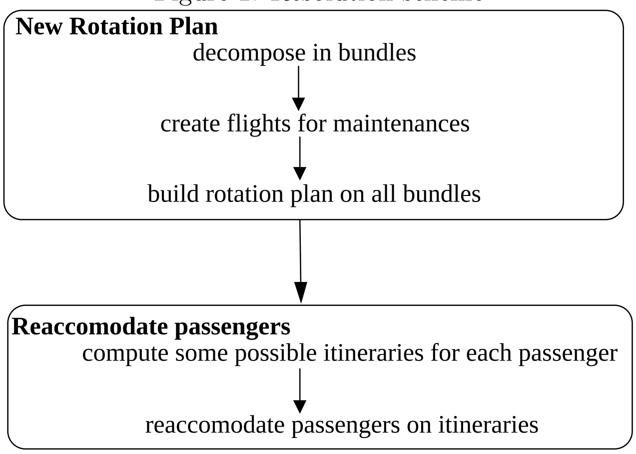
3 Algorithmic approach

3.1 Solution framework

Our main proposal is an algorithm by construction, which builds a feasible rotation plan and a feasible passenger re-accommodation and minimizes both operational and functional costs. The algorithm is based on a decomposition of the problem into two main steps. The first step concerns aircraft rotations and the second concerns passengers re-accommodation.
In order to obtain a solution of good quality, both the construction of the rotation plan and the re-accommodation of passengers, use the initial planning. So the solution mimics the previous solution, and both operational and functional costs are limited.
The simplicity of the algorithm used permits also to highlight their high potential. In fact, for both steps we use simple solution methods based on flow formulations. This also allows to take directly all constraints into account, so there is no need for any correction procedure.

Figure 1 gives an overview of the solution scheme. The subsequent description gives some precisions for each step.

Figure 1: Resolution scheme



New Rotation plan.
First we add the delays or cancel the initial disrupted flights. Then we identify the fixed flights, which are scheduled before the beginning of the recovery period in the initial program. We are not allowed to modify anything on these flights. From now on, we focus on non fixed flights.

We want to reassign aircrafts to these non fixed flights in order to build a new rotation plan. We have the guarantee that a solution respecting all constraints except the maintenance constraints exists. Indeed, one of those solutions is to cancel all non fixed flights. In simple cases, a feasible rotation plan with respect to maintenance constraints exists using only these flights. For more tricky instances, new flights are needed to bring aircrafts to their maintenance location. This is detailed in section 3.2.3.
In order to speed up and ease the construction of the rotation plan and the satisfaction of the maintenance constraints, we partition the problem into several bundles (section 3.2.2). A bundle is made on one hand from some aircrafts and on the other hand from different flights. We then

reassign aircrafts to flights in each bundle in a sequential way. Reassignment of aircrafts to flights is generic and is detailed in the section 3.2.1.

Passenger re-accommodation.
In this step, we re-accommodate passengers on operated flights. We first compute alternative itineraries for the passengers. Then we solve a path based multicommodity flow problem with capacity constraints in order to re-accommodate the passengers on the computed itineraries. Further algorithmic details are given in next sections.

3.2 New rotation plan

3.2.1 General scheme and model

The reassignment of aircrafts to the flights is done by using a generic method. This method is explained in this section, its aim is to provide a reassignment of some aircrafts, to some flights with respect to aircraft constraints and with a good hope for a low global cost. More precisely, input, constraints and output are as follows:

- INPUT:
  - a set K of aircrafts with same transit times and of the same family (for example, of the same model) and their availability airport AD^k and time TD^k,
  - a set F of flights (rotations) to these aircrafts are assigned,
  - a set D of possible delays to delay further the rotations,
  - a cost function c^k_{fd} which gives the cost of assigning aircraft k on flight f delayed by delay d.
- CONSTRAINTS:
  - respect of aircrafts' transit times,
  - for every required maintenance, bring aircraft to its maintenance on time, and the total flight time before required maintenance of aircraft k must not exceed $MaxFlightHours^k$,
  - do not use unavailable aircrafts (unavailability disruption),
  - respect airports take off and landing capacities.
- OUTPUT: a rotation plan respecting all constraints and minimizing the costs with respect to c^k_{fd} cost function.

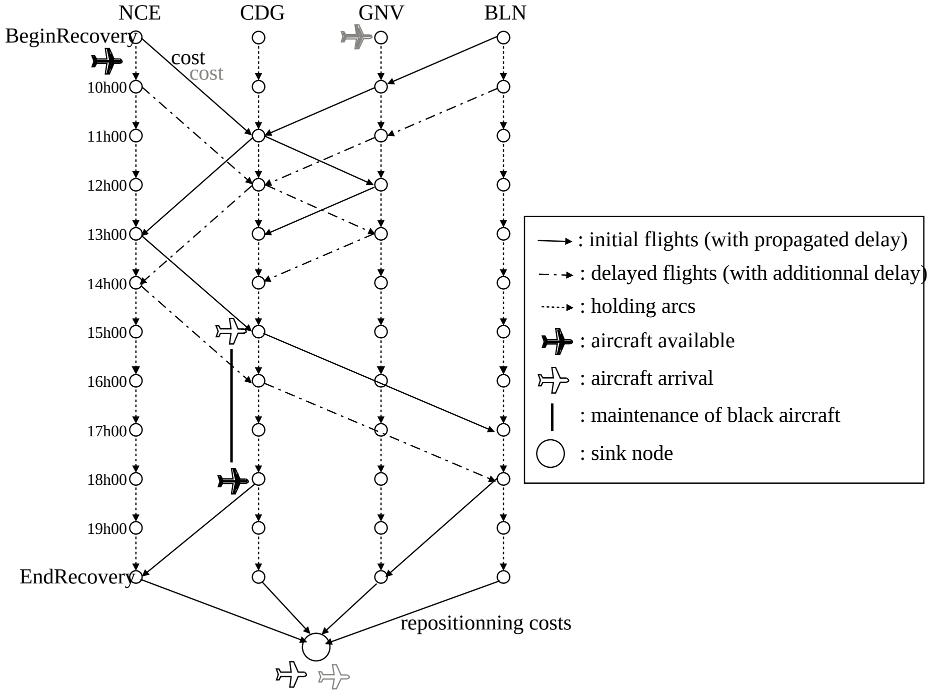
To capture and take advantage of the network structure, we use a mixed integer multicommodity network flow problem formulation as a basis for the mathematical model. We add some constraints in order to take the specific problem characteristics into account.

This research will use a time-space diagram as a graphic representation of the model (see Figure 2). Time is discretized with a step of Δ_t time units.

Each node within the network represents an event taking place in a specific airport at a specific time. We introduce several different types of nodes inside the time-space network as follows:

1. Classical node: most nodes represent airports at a specific time.
2. Aircraft availability nodes: Aircrafts are available for reassignment from a certain airport at a given time during the recovery period.
3. Aircraft arrival nodes: Maintenance requirements introduce some specific arrival nodes for some aircrafts. In example (figure 2), the black aircraft has to be at 15h00 at its maintenance location *CDG*.
4. Sink node: An artificial sink node is added at the end of the recovery as destination for all aircrafts.

Figure 2: Graphical representation of aircraft reassignment.



Each arc within the network shows the linkage among different nodes. The arcs can be represented as follows:

1. Flight arc: A flight arc connects aircraft departure node to aircraft arrival node, which means a possible flight. Transit times are included in flight durations, so two flights can be operated consecutively.

2. Ground holding arc: This arc connects both arcs within the same airport but different time horizon. The amount of flow in the arc represents the number of aircraft kept during the ground holding time. In our problem there are neither capacities, costs nor revenues on these arcs.

3. Delay arc: Delay arc is the production of the delay strategy. We add up some parallel arcs to the scheduled flight arc with some delays chosen in the delay set D, such as 5 or 60 minutes.

The problem remains at computing a minimal cost network flow with side constraints in this time-space network.
The **objective** is to minimize the costs induced by cancelled flights $c_f X_f$,and the costs induced by delayed flights or by changes of aircraft operating a flight $c^k_{fd} x^k_{fd}$.

$$\text{Minimize} : \quad \sum_{\substack{k \in K \\ f \in F, d \in D}} c^k_{fd} x^k_{fd} + c_f X_f, \tag{1}$$

Constraints (3) and (4) are the classical **flow conservation constraints**, and conditions at aircraft availability nodes and at the sink node. Constraint (2) allows to operate each flight no more than once, either on time or delayed.

$$\sum_{k\in K}\sum_{\substack{d\in D\\ TA_f+d\leq TE}} x^k_{fd} \leq 1,\ \forall f\in F, \tag{2}$$

$$\sum_{\substack{f\in F,\ d\in D\\ AA_f=j\\ TA_f+d=t-\tau^k}} x^k_{fd} + y^k_{j,t-\Delta_t} - \sum_{\substack{f\in F,\ d\in D\\ AD_f=j\\ TD_f+d=t}} x^k_{fd} - y^k_{j,t} = \begin{cases} -1 & if \quad j=AD^k, \\ 0 & else \end{cases} \tag{3}$$

$$\forall j\in A \quad , \quad \forall k\in K, \forall t\leq TE-\tau^k$$

$$\sum_{k\in K}\sum_{\substack{t\in T\\ t\leq\tau^k}}\sum_{\substack{f\in F,\ d\in D\\ AA_f=j\\ TA_f+d=TE-\tau^k+t}} x^k_{fd} + \sum_{k\in K} y^k_{f,TE-\Delta_t} \geq |K| \tag{4}$$

Airport capacity constraints (5) and (6) link different nodes and flights. With limited capacities, a choice has to be made. These linking constraints enforce the difficulty of solving the problem.

$$\sum_{k\in K}\sum_{\substack{f\in F,\ d\in D\\ AD_f=i\\ h\leq TD_f+d<h+1}} x^k_{fd} \leq \mathit{CapTakeOff}[i,h], \qquad \forall i\in A, \forall h\in H, \tag{5}$$

$$\sum_{k\in K}\sum_{\substack{f\in F,\ d\in D\\ AA_f=j\\ h\leq TA_f+d<h+1}} x^k_{fd} \leq \mathit{CapLanding}[j,h], \qquad \forall j\in A, \forall h\in H, \tag{6}$$

Maintenance constraints (7) and (8) are twofold. On the one hand, they guarantee that an aircraft stays at its maintenance airport during the maintenance. On the other hand, they limit the flight duration before the maintenance to $\mathit{MaxFlightHours}^k$.

Unavailability constraints (9) guarantee that no flight is done by an aircraft during its unavailability, in other words, aircraft stays in some airport. All these constraints are added to the problem.

$$y^k_{AM^k t} = 1, \qquad \forall k\in KM, \forall t\in[TMB;TME], \tag{7}$$

$$\sum_{\substack{f\in F,\ d\in D\\ TA_f+d\leq TMB^k}} \big(TA_f - TD_f\big) x^k_{fd} \leq \mathit{MaxFlightHours}^k, \qquad \forall k\in KM, \tag{8}$$

$$\sum_{i\in A} y^k_{it} = 1, \qquad \forall k\in KU, \forall t\in[TUB;TUE], \tag{9}$$

All **variables** are binary.

$$x^k_{fd}\in\{0,1\}, \qquad \forall k\in K, \forall f\in F, \forall d\in D, \tag{10}$$

$$X_f = 1 - \sum_{\substack{k\in K,\\ d\in D}} x^k_{fd} \in\{0,1\}, \qquad \forall f\in F, \tag{11}$$

$$y^k_{jt}\in\{0,1\}, \qquad \forall k\in K, \forall j\in A, \forall t\in T. \tag{12}$$

Discretization of time leads to huge models, consequently we need to decompose further the problem.

3.2.2 Decomposition in bundles

The size of instances varies from 608 up to 2844 rotations and from 85 up to 255 aircrafts. Moreover, an aircraft change can only be done within the same aircraft family. Thus it seems quite natural to decompose the construction of the new rotation plan, by bundles of aircrafts of the same family and flights operated by these aircrafts. This first decomposition guarantees that no change of family of aircrafts is done.

Even though, after this decomposition, for the largest instances some bundles remain too large. For example, on instance $B10$, the bundle of aircrafts of family *AirbusSmall* contains 144 aircrafts. Thus, we need a finer decomposition, and we make bundles of aircrafts of the same model. Moreover, this guarantees that all aircrafts have the same transit-time, this simplifies our algorithm. Some bundles, still remain too big (for instance $B10$, there are 60 aircrafts of model $A320$ and 1986 flights assigned to these aircrafts) and a reconstruction of the rotation plan on these bundles cannot be done in less than 1 minute. That's why we decompose further each bundle, by tacking only some aircrafts of the same model, and their rotations to make a bundle.

We then reassign aircrafts to flights in each bundle, in a sequential way. We treat in priority bundles with aircrafts needing maintenance, such as enough airport capacities are available for these aircrafts. For these bundles some additional flights may be created (section 3.2.3) in order to bring aircrafts to their maintenance airport.

3.2.3 New flights for maintenance

We consider a bundle. If some aircrafts need maintenance, we create some flights to bring the aircrafts to their maintenance location. In theory, choosing these flights can be very difficult, and tricky instances could show the limits of most simple heuristics. In fact, creating new flights depends on airports capacities. Thus we propose a simple algorithm (algorithm 1) of flight creation for maintenance. Its purpose is to create direct flights or flights with 1 stop-over to bring the aircraft from its availability airport to its maintenance airport.

Algorithm 1 Create new flights for maintenance

```
for aircraft k needing a maintenance (∈ KM) do
  Calculate AD^k and TD^k ;
  Read AM^k , TMB^k and MaxFlightHours^k;
  if Dist(AD^k,AM^k) ≤ MaxDist^k and available airport capacities) then
    create a direct flight from AD^k to AM^k as early as possible and arriving before TMB^k;
  else
    for airport a ∈ A do
      create (if possible) a direct flight f from AD^k to a as early as possible;
      create (if possible) a direct flight f′ from a to AM^k as early as possible after f;
    end for
  end if
end for
```

3.2.4 Reassignment costs

The cost function is computed in order to get a similar rotation plan as the initial plan. Thus it may allow a good passenger re-accommodation and limits passengers inconvenience. So, it is of first importance to have a precise cost function, which includes both, operational and functional costs.

The cancel cost c_f of flight f is the cancel cost of all passengers previously assigned to flight f. This includes the prices paid by the passengers, the compensation cost and some penalties.

The cost c^k_{fd} of assigning aircraft k to flight f delayed by delay d includes the cost of a change of aircraft operating flight f and the delay cost. A change of aircraft may imply a cabin class change or cancelling flights of some passengers if the assigned aircraft is smaller. This is reflected in the cost.

Ideally, this cost has also to reflect the impact of passengers with correspondences. In fact, these passengers with one or more transits represent an important part of the revenues. We did not find a satisfactory linear approximation of these costs, thus we could not include them in our formulation.

3.3 Passenger re-accommodation

In this step we re-accommodate passengers on operated flights. We first compute alternative itineraries for the passengers. Then we solve a path based multicommodity flow problem with capacity constraints in order to re-accommodate the passengers on the computed itineraries.

3.3.1 Itineraries for re-accommodation

For each passenger p we compute a set Q^p of alternative itineraries with respect to the re-accommodation constraints: an alternative itinerary has the same origin and the same destination as the initial itinerary, departure time of alternative itinerary is later than the initial departure time, and the delay at arrival does not exceed $MaxDelayPassenger^p$. These itineraries include the different cabin classes (F/B/E). Simultaneously, the cost C_q^p of assigning one passenger of p, to the alternative itinerary q is computed. This cost includes the downgrading costs, the delay costs and the inconvenience costs.
The algorithm for the computation of alternative itineraries for passengers and costs is given in algorithm 2.

Algorithm 2 Compute alternative itineraries for passengers

for passengers p in P **do**
  Read AD^p, AA^p, TD^p, TA^p and transit airports;
  Compute alternative itineraries from AD^p to AA^p (depth first search);
  Assign all combinations of possible cabin classes to these itineraries;
  Compute the costs C_q^p;
  Add them to Q^p;
  for a in A, a hub **do**
    Compute alternative itineraries from AD^p to AA^p via hub a;
    Assign all combinations of possible cabin classes to these itineraries;
    Compute the costs C_q^p;
    Add them to Q^p;
  end for
end for

In our study, we compute these alternative itineraries statically and once for all. They are then used to re-accommodate passengers on them, via a path based multicommodity flow problem with capacity constraints which is presented in the next section.

3.3.2 Model

We denote by z_q^p the integer variable representing the number of passengers belonging to passengers p, assigned on itinerary q, C_q^p is the corresponding reassignment cost. The total number of passengers of p is n^p. If a passenger of p is not re-accommodated, his trip is cancelled, this costs CC^p.
δ_{fq}^{kcl} is boolean, of value 1 when flight f operated by aircraft k considering cabin class cl is in itinerary q, else its value is 0. $cap^k(cl)$ is the seat capacity in cabin class cl of aircraft k.
The model we solve is:

$$\text{Minimize}: \quad \sum_{p\in P}\sum_{q\in Q^p} C^p_q z^p_q + \sum_p CC^p\Big(n^p - \sum_{p\in Q^p} z^p_q\Big), \tag{13}$$

s.t.

$$\sum_{p\in P}\sum_{q\in Q^p} \delta^{kcl}_{fq} z^p_q \le cap^k(cl), \qquad \forall f\in F,\ \forall cl\in CL, \tag{14}$$

$$\sum_{q\in Q^p} z^p_q \le n^p, \qquad \forall p\in P, \tag{15}$$

$$z^p_q \in N. \tag{16}$$

Objective function (13) contains two parts, the first concerns the re-accommodation costs and the second concerns the cancel cost of non re-accommodated passengers. Constraint (14) guarantees that the seat capacity of all aircraft is respected in each cabin class. And constraint (15) guarantees that the total number of re-accommodated passengers of p does not exceed the initial number of passengers of p. All other functional constraints are already included in the itinerary generation.

In theory, by generating all possible alternative itineraries, this model gives the optimal re-accommodation of all passengers on operated rotations. Of course, in practice, we only generated some good alternative itineraries as detailed in the previous section. A better way is to implement a column generation algorithm, which generates dynamically alternative itineraries if needed to re-accommodate passengers. Moreover, its quality depends on the operated rotations computed in the first step. Thus a feedback on rotation construction should be done to improve the rotation plan with regard to passengers.

4 Computational study

In this section we present our computational results on the instances used to establish the final rank of the participants of challenge ROADEF 2009. The instances are divided in three sets: instances A were used during the qualification phase and were available from the beginning. Instances B were available after the qualification, whereas the instances XA and XB were unknown until the end of the challenge.

The evaluation is made on instances B, XA and XB (18 different instances) composed of 2844 rotations operated by 255 aircrafts and about $11,000$ itineraries which correspond to more than $200,000$ passengers. The solution on each instance must be computed in less than 10 minutes on a Turion64x2 with 2GB of RAM.

The computation time of each part of the algorithm is detailed in Table 1. For each instance, we detail the percentage of time spent on each function: Construction of the new rotation plan which concerns *aircrafts* an re-accommodation of passengers which concerns *passengers*. We refer to these two functions as *aircrafts* and *passengers* in Table 1. The last column presents the total time (in seconds) needed to obtain the solution. The loading time of the instance and the writing time of the solution are omitted since they take less than a few seconds. Table 1 shows that about 20% of the computation time is spent on building the new rotation plan and 80% on the re-accommodation of the passengers. On small instances such as *XA01* and *XA03*, the passenger re-accommodation is done in less than 10 seconds and most of the computation time is spent on the new rotation plan. On the bigger instances, repartition of time is opposite. This can be explained by the size of the mathematical program used in each function. The construction of the new rotation plan is done by solving several small mixed integer programs whereas the passenger re-accommodation solves only one large integer program.

The score on each instance is computed by dividing the *deviation* by the *range*. The *deviation* is the difference between the value of the solution and the worst solution, while the *range* is the

Table 1: Detailed computation time

Instance	Aircrafts(%)	Passengers (%)	Total time (s)
B01	17.71	82.29	336.985
B02	19.68	80.32	302.641
B03	18.17	81.83	329.109
B04	18.25	81.75	322.672
B05	19.89	80.11	465.578
B06	19.49	80.51	367.188
B07	19.95	80.05	357.782
B08	18.04	81.96	393.594
B09	17.44	82.56	406.203
B10	19.38	80.62	483.141
XA01	93.07	6.93	18.032
XA02	29.74	70.26	219.000
XA03	93.21	6.79	18.656
XA04	26.68	73.32	191.781
XB01	18.51	81.49	320.750
XB02	19.9	80.1	466.797
XB03	20.11	79.89	353.875
XB04	19.82	80.18	475.250

Table 2: Computational results

Instance	Best	Worst	Result	Score
B01	971 182.50	43 169 547.75	2 480 629.85	0.96
B02	1 220 708.30	31 420 941.20	5 741 677.20	0.85
B03	1 007 565.70	47 509 155.15	2 579 411.00	0.97
B04	1 101 394.80	46 400 734.65	2 575 466.95	0.97
B05	9 653 780.05	94 278 109.15	29 235 529.65	0.77
B06	3 218 000.10	66 101 253.95	8 323 534.40	0.92
B07	5 039 744.20	56 785 261.80	16 318 727.85	0.78
B08	3 509 318.00	62 391 786.00	8 211 402.55	0.92
B09	3 967 344.70	68 668 311.00	10 101 024.65	0.91
B10	34 523 605.00	124 900 519.50	66 458 200.35	0.65
XA01	116 195.20	7 486 622.70	264 756.30	0.98
XA02	1 475 322.10	38 313 615.30	3 582 627.75	0.94
XA03	285 287.05	10 092 302.00	604 065.45	0.97
XA04	4 112 262.60	13 937 165.80	10 487 502.00	0.35
XB01	1 352 823.05	105 894 332.10	3 156 666.95	0.98
XB02	11 297 822.20	200 281 943.50	29 259 788.40	0.90
XB03	6 463 354.30	135 863 963.60	12 609 345.75	0.95
XB04	34 331 225.80	240 102 702.80	63 522 137.85	0.86
Mean				0.87

difference between the best and the worst solutions. The best solution is the minimum total cost found among all participants whereas the worst solution is the maximum total cost. The participant score is the average of their score on each instance.

A slight modification of the code provided for the final step allows us to have feasible solutions on all instances. This is done by simply adding a new constraint into our model without modifying the other parts of the code. Table 2 shows the value of the best solution, the worst solution and the total cost of our solutions. The last column is the score previously defined. Using the challenge ranking system, the new version of our algorithm provides the third best result with an average score of 87%! The version provided for final ranking, provided the fifth best result, with an average score of only 70.62% reflecting some unfeasible solutions.

So the proposed algorithm provides not only feasible solutions on all instances, but high quality solutions. This shows the potential of simple techniques based on multicommodity flows. Moreover, the simplicity of the proposed algorithm enables to easily include new constraints in the problem.

5 Conclusions and perspectives

In this paper we studied the airline disruption management problem which was the topic of the ROADEF'09 challenge. We developed a 2-step decomposition heuristic to solve the problem. We modelled both of the steps as multicommodity network flow problems with side constraints, respectively with arc and path formulations.

Thus, we tackled this problem by solving sequentially multicommodity network flow problems with side constraints with a MIP solver. The method is not only simple, but it also guarantees that a feasible solution is built from the beginning. Thus there is no need of any correction procedure. Both these aspects make the originality of our contribution.

The perspectives are twofold. First, our decomposition scheme may be enhanced by improving each step; in particular a column generation scheme would improve and accelerate passenger re-accommodation. Second, a feedback after re-accommodation of passengers on aircraft assignment may allow improving gradually the obtained solution. Furthermore, it would integrate even more the needs of passengers in aircraft re-assignment.

References

[1] T. Andersson. The flight perturbation problem: operational aircraft rescheduling. 2001.

[2] T. Andersson. Solving the flight perturbation problem with meta heuristics. *Journal of Heuristics*, 12(1):37–53, 2006.

[3] T. Andersson and P. Värbrand. The flight perturbation problem. *Transportation Planning and Technology*, 27(2):91–117, 2004.

[4] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9(3):279–298, 2006.

[5] J. Clausen, A. Larsen, and J. Larsen. *Disruption management in the airline industry-concepts, models and methods.* IMM, Informatik og Matematisk Modelling, DTU.

[6] A.I.Z. Jarrah, G. Yu, N. Krishnamurthy, and A. Rakshit. A decision support framework for airline flight cancellations and delays. *Transportation Science*, 27(3):266, 1993.

[7] J.G. Klincewicz and M.B. Rosenwein. The airline exception scheduling problem. *Transportation Science*, 29(1):4, 1995.

[8] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline disruption management-Perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162, 2007.

[9] L. Lettovsky. Airline operations recovery: An optimization approach. 1997.

[10] M. Love, KR Sorensen, J. Larsen, and J. Clausen. Disruption management for an airline-rescheduling of aircraft. *Lecture Notes in Computer Science*, pages 315–324, 2002.

[11] M. Palpant, M. Boudia, C.-A. Robelin, S. Gabteni, and F. Laburthe. Roadef 2009 challenge: Disruption management for commercial aviation, 2008.

[12] J.M. Rosenberger, E.L. Johnson, and G.L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408, 2003.

[13] D. Teodorović and G. Stojković. Model to reduce airline schedule disturbances. *Journal of Transportation Engineering*, 121:324, 1995.

Les cahiers Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication sur des problèmes liés au mathématiques discrètes.

Responsables de la publication : Nadia Brauner et András Sebő
ISSN : 1298-020X -