ROADEF 2009 Challenge
Description of the TUe Solution Method

Christian Eggermont, Murat Firat, Cor Hurkens, Maciej Modelski

Eindhoven University of Technology, The Netherlands

1 Introduction

The complex problem presented in [1] is decomposed into smaller more tractable pieces. Each subroutine corresponding to a sub-problem improves the current schedule in one aspect and its output is input for the next one.

The subproblems we identified deal with fixing the aircraft rotation continuity, respecting the airport capacity constraints, fine-tuning delay management, and itinerary reassignment. Each of the subproblems is described in some more detail below.

2 Fixing Aircraft Rotations

Due to flight cancellations and aircraft unavailability periods it may happen that an aircraft's rotation gets broken at some point. We start by canceling for an unavailable aircraft all remaining flights originally scheduled within its unavailability period.

Next we list for each aircraft its assigned (non-canceled) flights in order of (planned) departure time. We treat consecutive flights of aircraft in pairs and cancel and/or add flights if certain conditions are satisfied. Assume we are considering a flight $A \to B$ followed by $C \to D$, where $C \neq B$. If $D = B$ the problem is resolved by canceling the second flight. Otherwise we create an additional flight $B \to C$, or a flight $B \to D$ while canceling $C \to D$.

After this first fix, each flight is preliminarily scheduled to depart at the first available moment in time, not necessarily respecting airport arrival and departure capacities.

3 Respecting airport capacities

For each airport and each time slot of an hour, the number of arrivals and the number of departures are restricted. If we handle aircraft one-by-one, and schedule each flight as early as possible, while respecting airport capacities, we may run into the problem that either an aircraft arrives too late for maintenance or it cannot carry out all its assigned flights within the recovery period. In these cases we have to cancel or exchange some of its flights.

In order to fix this problem, we first discriminate between flights leading towards a maintenance period, and those that do not. For an aircraft that goes towards maintenance, the first set of flights constitute a so-called *pre-maintenance rotation*. The remainder of the flights per aircraft form a *non-pre-maintenance rotation*. As maintenance due dates have to be respected, we schedule pre-maintenance rotations first, aircraft by aircraft, aircraft with earliest maintenance first. Next we schedule the remaining, non-pre-maintenance rotations, aircraft by aircraft, with aircraft in random order.

For each rotation, we start scheduling flights at the earliest possible time as long as the airport capacities allow. If this works out, we fix the departure times and adjust remaining airport capacities. If it does not work, we compute for each flight in the rotation a latest possible departure time. Based on these earliest and latest departures we decide how to short-cut the rotation by one of three ways : either skip a middle section of flights, starting and ending at the same airport ; or skip a trailing segment of flights, ending up at the wrong airport ; or skipping a middle section of flight, while adding a flight so as to connect the first section to the last section.

4 Delay fine-tuning

After constructing feasible rotations in terms of continuity, airport capacity, turn around or transit times between consecutive flights of aircraft, and maintenance due dates, we may want to have additional delays to have enough connection time for as many passengers as possible.

Flights may be subjected to delays in a way that they stay in their original time slots. Let Δ_i denote the maximum delay for which both departure and arrival of flight i stay in their respective time slots.

Given the current schedule we need not consider connections that are surely lost, and neither connections that are certainly made. For the remaining connections the objective tries to achieve maximal slack. To this purpose we add to the objective a term $w_{ij}(X_j - X_i)$, where variables X_k refer to departure times of flight k, and the weight w_{ij} counts the number of passengers hoping for a feasible connection from flight i to flight j. The variables X_k are to stay within their range $[\hat{X}_k, \hat{X}_k + \Delta_k]$, where $\hat{X}_k$ is the current planned departure time for flight k.

5 Itinerary reassignment

With flight cancellations and delays introduced both by the problem instance and the preceding steps of our algorithm, many itineraries become infeasible. Here we try to reroute passengers of an itinerary using the available capacity of the operated flights.

We solve the problem separately for each itinerary by finding a minimum cost $s-t$ flow in the graph depicted in Figure 1. We construct this graph once and adjust it for each problematic itinerary under consideration.

By assigning appropriate capacities and costs to the arcs in the graph, we find for a particular problematic itinerary the cheapest feasible alternative routes. Those passengers that cannot be accommodated, are canceled.

Références

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

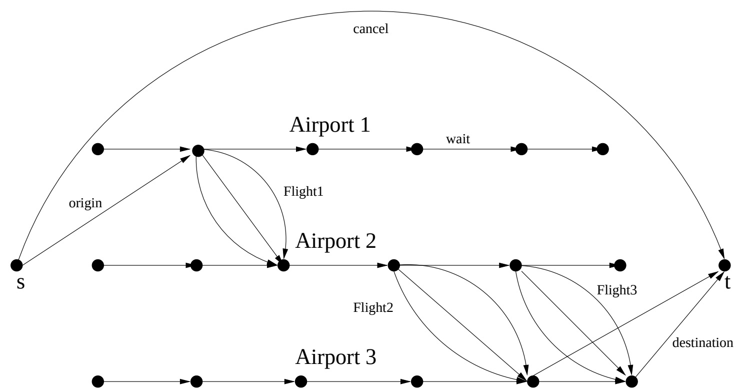

Fig. 1. The flow of itineraries.