A New Methodology to Solve the Stochastic Aircraft Recovery Problem using Optimization and Simulation

Pol ARIAS [a], Daniel GUIMARANS [b] and Miguel MUJICA [a,c]
[a] *Universitat Autonoma de Barcelona (UAB), Sabadell, Spain*
[b] *National ICT Australia (NICTA), Sydney, Australia*
[c] *Amsterdam University of Applied Sciences, Amsterdam, The Netherlands*

Abstract. The Aircraft Recovery Problem (ARP) appears when external events cause disruptions in a flight schedule. Thus in order to minimize the losses caused by the externalities, aircraft must be reallocated (rescheduled) in the best possible way. If uncertain conditions are taken into account the Stochastic Aircraft Recovery Problem (SARP) arises. The aim of this paper is to develop a suitable approach based on Constraint Programming paradigm and using simulation to solve this so-called SARP . The approach solves the problem through the rescheduling of the flight plan using delays and swaps. The main objective is to restore as much as possible the original flight schedule, minimizing the total delay. Several tests have been carried out on medium-sized scenarios to assess the accuracy of the solutions provided by our approach.

Keywords. Constraint Programming, Simulation, Aircraft Recovery Problem

1. Introduction

Operational disruptions are defined as a deviation from originally planned operations. The airline industry is notably one of the most affected industries regarding operational disruptions. The costs associated to them have gained more and more importance with the increase of fuel costs and the punctuality policies that airlines have been forced to implement in order to maintain competitiveness [1]. Due to these and other emerging restrictions that the aeronautical industry is facing nowadays, the optimization of resources and time has become an important issue in the aeronautical agenda [2]. Moreover, a study developed by the airport of Gatwick [3] calculated that 30 % of the delays are caused by air traffic management (ATM). In addition, a 25 % of them are due to delays in land services, commonly known as the turnaround process

The Aircraft Recovery Problem (ARP) main objective is to restore the flight schedule as much as possible using the existing aircraft, i.e. minimize the total delay. Introducing some uncertainty in the values associated to the problem, i.e. flights duration or delays, the Stochastic Aircraft Recovery Problem (SARP) arises.

In this work we present a novel approach to tackle the SARP that combines optimization and simulation. First, given an original flight schedule and one or more disruptions (e.g. flight delays, airport closure, etc.), the solving approach based on Constraint Programming (CP) generates a solution through delaying and swapping aircraft to flights assignments, in order to create a feasible flight plan that minimizes the impact of the delays as much as possible. Such plan considers all flights scheduled within a certain period of time by a given fleet including the original departure, the expected flight durations, and the connections between airports.

The simulation approach is used to generate variations of the original scenario according to this stochasticity, in order to evaluate the robustness of the solution provided by the optimization method under uncertain conditions. Recent studies analyse the robustness of the final re-schedule [4] in contrast with previous airlines priorities of just minimizing total delays. The main argument is that in networks with a large number of connecting flights, delays can propagate very rapidly throughout the scenario. This increases the recovery costs of the airlines and has a larger impact on their profit.

The literature contains several works on different aspects of the ARP. As mentioned, a recent work by Dunbar et al. [4] is focused on the robustness of the solution by integrating aircraft routing and crew pairing. Lan et al. [5] develop a robust aircraft routing model to minimize the expected propagated delay along aircraft routes. They use an approximate delay distribution to model the delay propagation and use a branch-and-bound technique to solve their MIP. Instead of estimating delay propagation, Wu [6] used a simulation model to calculate random ground operational delays and airborne delays in an airline network. Wu [6][7] shows that delays are inherent in airline operations due to stochastic delay causes.

The article is organized as follows. Section 2 presents the CP formulation. Section 3 introduces the approach developed for the stochastic ARP. In section 4 some tests are presented. Finally, in Section 5, the conclusions are given and lines for future research are outlined.

2. Constraint Programming Formulation

The optimization method is based on the Constraint Programming (CP) paradigm [8], specially suitable for problems regarding scheduling, routing, planning, and resource configuration. CP is based in three entities: (i) variables, (ii) their corresponding domains, and (iii) the constraints relating these variables. The main solving techinque is known as *constraint propagation* [9]. There are artifices to increase the efficiency of this technique; one of them is the addition of redundant constraints to further prune the search tree. We take advantage of this characteristic in our formulation of the ARP, thus modelling the problem with two sets of variables: predecessors (P) and successors (S). These variables allow us modelling the same search space from two different perspectives, while the redundant constraints propagate decisions made in any of the two sets to the other one. This formulation is inspired on the Vehicle Routing Problem formulation by Kilby and Shaw [10].

We consider a set of n flights and a fleet of m aircraft. Then, the variables used in this formulation are:

- $\Psi = \psi_1 ... \psi_n$ are the flights to be attended;
- $A = a_1 ... a_m$ are the available aircraft;
- $G = g_1 ... g_{n+2m}$ are the assignment set, with domain $G :: [1..m]$.

It should be noticed that there is one assignation per each flight and two special assignations per aircraft: the starting and ending airports for the aircraft. Thus, two subsets of G, F and L, are defined as the aircraft departure and arrival airports to ensure the closure of the cycle:

- $F = n+1...n+m$ is the set of first assignments;
- $L = n+m+1...n+2m$ is the set of last assignments.

Then, the predecessor and successor sets are defined as:

- $P = p_1...p_{n+m}$ is the predecessors set, with domain $P :: [1..n+m] :: (G-L)$;
- $S = s_1...s_{n+m}$ is the successors set, with domain $S :: [1..n, n+m+1..n+2m] :: (G-F)$.

A set of constraints is imposed to relate all the variables and define the problem. The predecessor and successor variables form a permutation of G and are therefore subject to the *difference constraints* (1).

$$p_i \neq p_j \quad \forall i, j \in G \wedge i < j \qquad s_i \neq s_j \quad \forall i, j \in G \wedge i < j \tag{1}$$

These equations force predecessor and successor sets to contain no repetitions. Thus, one flight can have one and only one predecessor and successor.

The successor variables are kept consistent with the predecessor variables via the following *coherence constraints*:

$$s_{p_i} = i \quad \forall i \in G-F \qquad p_{s_i} = i \quad \forall i \in G-L \tag{2}$$

Equations (2) connect the concepts successor and predecessor as follows: the former shows that i is the successor of its predecessor, and the latter indicates that i is the predecessor of its successor.

Along a set of connected flights, all assignations are performed by the same aircraft. This is maintained by the following *leg constraints*:

$$g_i = g_{p_i} \quad \forall i \in G-F \qquad g_i = g_{s_i} \quad \forall i \in G-L \tag{3}$$

Equations (3) are used to ensure that the aircraft assigned to i is the same as that assigned to its predecessor and successor.

Other sets of variables are defined to ensure the connections between origin and destination airports, as well as the times that aircraft are assigned to their flights:

- $O = o_1...o_n$ is the origin airport set;
- $D = d_1...d_n$ is the destination airport set;
- $\Delta = \delta_1...\delta_n$ is the duration time list;
- $T = t_1...t_n$ is the departing times list, indicating the time when the aircraft i departs;
- $S = s_1...s_n$ is the scheduled times list, indicating the time when the aircraft is originally scheduled to depart;
- $\Gamma = \gamma_1...\gamma_n$ is the initial delays list, indicating the delays that have disrupted the system;
- $\Lambda = \lambda_1...\lambda_n$ is the delays list, indicating the accumulated delays of each flight.

The real departure time is calculated given the *departure time constraints*:

$$t_i \geq t_{p_i} + \delta_{p_i} \quad \forall i \in G - F \qquad t_i \leq t_{s_i} - \delta_i \quad \forall i \in G - L \tag{4}$$

Equations (4) bound the real departure time of an aircraft assigned to the flight i. This time is, at least, the departure time in the predecessor of i, plus the duration time from the predecessor to i (δ_{p_i}). Equally, this time must be, at most, the departure time in the successor, minus the duration time from i to its successor (δ_i).

The connection between origin and destination airports is done by using the *connectivity constraints*:

$$o_i = d_{p_i} \quad \forall i \in G - F \qquad d_i = o_{s_i} \quad \forall i \in G - L \tag{5}$$

Equations (5) are used to narrow down the combinations of flights. The origin of flight i must be the destination of its predecessor. In the same way, the destination of flight i is the origin of its successor.

Equation (6), ensures that the departing time of i is greater than the scheduled time plus the initial delay.

$$t_i \geq s_i + \gamma_i \quad \forall i \in G - F - L \tag{6}$$

Equation (7) allows to calculate the total accumulated delay by obtaining the difference between the real time of departure (t_i) and the scheduled time of the departure (s_i).

$$\lambda_i = t_i - s_i \quad \forall i \in G - F - L \tag{7}$$

Finally, the objective function (8) to be minimized is defined as the sum of accumulated delays for all flights.

$$\min \sum_{i=1}^{n} \lambda_i \tag{8}$$

3. Methodology

As explained before, the ARP is very unlikely to occur in the deterministic form. As a lot of tasks have to be done to ensure that an aircraft is ready to depart, many sources may introduce uncertainty to the system. Therefore, the nature of the problem may be considered stochastic. The resulting Stochastic ARP (SARP) becomes more challenging than the original ARP.

In Figure 1 a diagram is introduced to outline the proposed methodology. First, the stochastic problem is simplified to a deterministic instance by using the average values of the adjusted probability distributions of the different processes. An initial solution is generated using the original flight assignment. This solution provides an initial value for the total delay (the cost function), which is used as an upper bound for the objective function in the local search performed by the CP approach. A better flight schedule reducing the total delay is found as the result of this process.

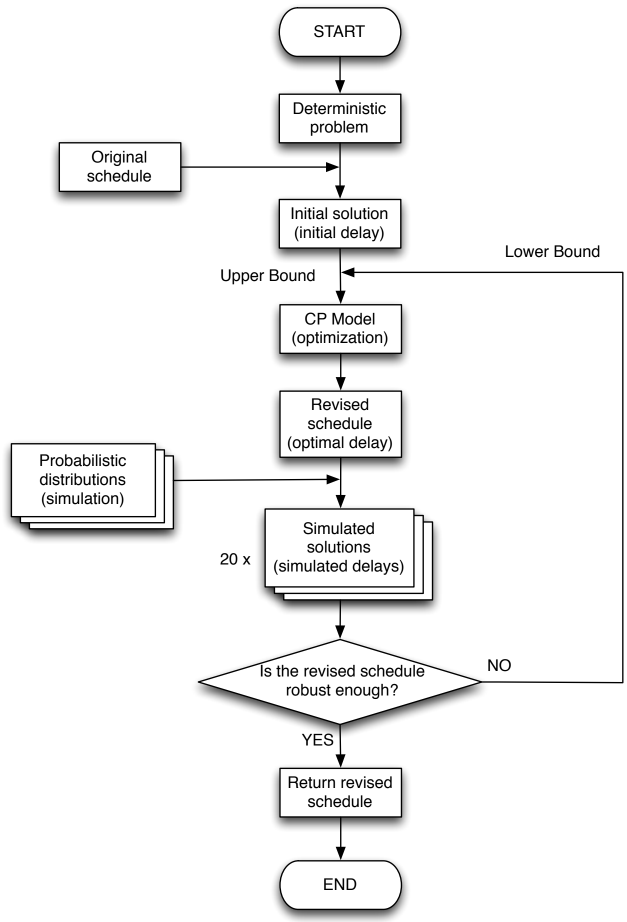

Figure 1. Methodology flow diagram

This optimized solution is then checked using simulation to verify its robustness: a set of 20 stochastic instances is generated using the probability distributions for the processes. Maintaining the improved flight schedule returned by the CP model, we compute the total delay for each instance. This way, a single solution is evaluated in 20 different scenarios. The results are then analysed in order to determine the level of robustness of the obtained solution. At this stage, different criteria can be considered to determine whether a solution is robust or not. First, a solution may be considered to be robust if the standard deviation of the simulated solutions is proportional to the variation of the used probabilistic distributions and its expected propagation due to problems size. Second, a solution may be considered robust if the gap between the average of the simulated solutions and the deterministic solution falls within a tolerance interval. Third, we may define the criterion as the number of solutions whose gap to the deterministic solution is smaller than a given value. Finally, operational considerations such as the number of swapped flights / aircraft assignments may be introduced. In the application case presented in this work, we use the first criterion to determine the robustness of the obtained solutions.

If the solution is not accepted as a robust one, its objective function value is used as a lower bound and the optimization/simulation process is repeated. This way a worst solution may be found but with a better robustness. Otherwise, the solution is accepted and the algorithm ends.

4. Application

To the best of our knowledge no defined benchmark instances exist neither for the ARP nor the SARP. For this reason, we defined a test instance composed by a total of 50 flights, 11 aircraft, and 10 airports. In order to simulate a local disruption, we introduce a delay of 120 minutes to the first 5 flights allocated in the first airport. In addition, some uncertainty is added to the duration of each flight. We simulate this fact by considering a normal probability distribution whose mean value is the estimated flight duration. The standard deviation is set to be 5 % of the flight duration. As no real data was available, a 5 % is chosen to illustrate enough variability during the flight duration. By choosing this deviation, we introduce some reasonable variation to our system. This permits validating the obtained results and the robustness of our solutions. According to this variation and the size of the defined scenario, we consider a solution to be robust if the standard deviation obtained from the simulated scenarios is less or equal to 5 % . These tests were conducted using the ECLiPSe CP platform [11] in an Intel Core i5 2.5 GHz with 4Gb RAM.

As a first step, we take the mean values of the normal distributions to obtain the deterministic version of the ARP. The total delay is calculated as the sum of all the delays present throughout the system. As can be observed in Table 1, applying the CP-based optimization process allows us to obtain a revised flight schedule for the new scenario, reducing the total delay in 26.34 %.

	Original flight schedule (min)	**Improved flight schedule (min)**	**Gap(%)**
Total delay	2050	1510	-26.34

Table 1. Deterministic results

Table 2 shows the results for the 20 stochastic instances generated using the simulation approach. For comparison, two obtained solutions are studied: the best solution found with a total delay of 1510 minutes, and a worse solution with a total delay of 1610 minutes. For each case, the corresponding solutions are reported, as well as the average and the standard deviation. The total delay is calculated by keeping the same flight assignation as in the deterministic case. Therefore, the variations in solutions' value are given by the deviations in the input data for each scenario, i.e. the variations in flights' duration. Even though, the first solution gives a tolerable standard deviation according to the chosen 5 %, the results of a worse solution (1610) are also reported. The purpose for testing the 1610 solution was to verify the robustness of solutions provided by the presented methodology. In this case, the best solution (1510) is accepted and returned by the method, as its standard deviation (4.22 %) falls within the defined tolerance gap (5 %).

Scenario	Total delay (min)	
	Solution 1510	**Solution 1610**
1	1616	1662
2	1564	1660
3	1614	1649
4	1599	1645
5	1583	1627
6	1546	1630
7	1502	1560
8	1556	1664
9	1539	1633
10	1526	1602
11	1586	1651
12	1530	1573
13	1479	1538
14	1459	1565
15	1423	1437
16	1523	1621
17	1608	1727
18	1524	1511
19	1639	1640
20	1702	1755
Average	*1548.2*	*1617.5*
St. Dev	*4.22 %*	*4.46 %*

Table 2. Stochastic results for the 20 simulated scenarios

5. Conclusions

This paper presents a methodology to solve the Stochastic Aircraft Recovery Problem (SARP). This methodology combines CP and simulation techniques. First, the CP model is used in a local search process to find an optimized solution. Next, simulation is used to check if the given solution is robust enough. Thus, the inherent stochasticity of the problem is naturally introduced in the decision making process. If the so-obtained solution does not achieve some imposed robustness criteria, it is discarded and a new optimized solution is generated.

The main contribution of this paper is a methodology combining optimization / simulation approaches where results are easily propagated between both techniques. The solutions obtained from the optimization method are easily perturbed and tested in the simulation scenarios. By applying this approach, significant results are obtained.

Finally, as an active field of research, this methodology is to be tested using real data scenarios. Moreover, the methodology may be extended to tackle more complex variants of the problem. For instance, a joint problem combining ARP characteristics with flight crews scheduling is under consideration.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] Zhang X, Zhao, M, Kuang, S M., and Du, Q. Research on Airline Company Fuel-Saving Model Based on Petri Network. Advanced Materials Research, 2013, 616, 1107-1110.

[2] Yu, G. Operations research in the airline industry . Kluwer Academic Pub, 1998, Vol. 9.

[3] ACI-Europe, Official News Bulletin for Members, Airport Council International Europe: Brussels, (2000).

[4] Dunbar, M., Froyland, G., and Wu, C. L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. Transportation Science, 46(2), 204- ?216.

[5] Yu, G. Operations research in the airline industry . Kluwer Academic Pub, 1998, Vol. 9.

[6] C. L. Wu. Inherent Delays and Operational Reliability of Airline Schedules. Air Transport Management, 11,273282, 2005.

[7] C. L. Wu,. Improving Airline Robustness and Operational Reliability by Sequential Optimisation Algorithms. Network and Spatial Economics, 6:235251, 2006.

[8] Rossi F, Van Beek P, Walsh T (Eds.), Handbook of constraint programming, 2006, Elsevier.

[9] Bessiere C, Constraint Propagation. Handbook of Constraint Programming, 2006, chapter 3, 29-83. Elsevier.

[10] Kilby, P. and Shaw, P. Vehicle Routing. Handbook of Constraint Programming, chapter 23, 801-836. Elsevier.

[11] Apt, K. and Wallace, M. Consrtaint Logic Programming using ECLiPSe. Cambridge University Press, 2007.