A Simple and Efficient Metaheuristic for the Dynamic Flight Scheduling Problem

Pol Arias[1], Daniel Guimarans[2], Angel A. Juan[3]

[1] Loughborough University
Epinal Way, Loughborough LE11 3TU, UK
p.arias@lboro.ac.uk

[2] Amsterdam University of Applied Sciences
Weesperzijde 190, 1097 DZ Amsterdam, Netherlands
d.guimarans.serrano@hva.nl

[3] Universitat Oberta de Catalunya
Av. Carl Friedrich Gauss, 08860 Castelldefels, Spain
ajuanp@uoc.es

Abstract

One of the biggest problems in the airline industry is facing the uncertainty created by passengers demand. In this work, we introduce a metaheuristic that uses updated demand forecasts to optimise a given flight schedule. The algorithm uses two main techniques to solve the problem: *(i) re-timing*, aimed at changing the scheduled time of a flight by delaying or advancing it to allow new connections; and, *(ii) re-fleeting* is focused on swapping aircraft between flights to increase aicraft's occupancy and better adjust it to demand. The algorithm's objective is to maximise airlines revenue by increasing the occupancy of each flight. This metaheuristic has been tested in different scenarios of up to 600 flights with promising results in low CPU times.

1 Introduction

Building a flight schedule is considered one of the key procedures in the airline industry [2]. Airlines need to deal with this problem in the early stages of the strategic planning horizon, when available information may still not be accurate enough. Decisions involved in this step, such as markets to serve, frequencies, capacity, and slots to offer, are often made between six months and one year before the flight departure. Therefore, plans made at this stage may be refined as more information is available, reducing inaccuracy and uncertainty. Because of these circumstances, using an algorithm that re-optimises and adapts a flight schedule to new forecast information will increase fitness to demand levels. In [1, 2], the authors show that, by using a dynamic flight schedule, a specific airline can improve its revenue by 2-2.5%.

The key element of this problem is the concept of *market*. A market is a pair of two airports (origin/destination), within a particular time frame. Every market has associated a certain demand. Passengers flying between these two airports can do so in a direct (non-stop) flight, or via an intermediate airport if suitable connections exist. The main objective of this work is to present a metaheuristic that increases the number of serviced passengers, adapting an existing schedule when more information is available; i.e. reducing the gap between the capacity of a market and its updated demand. Two main techniques are used in the algorithm: *re-timing* and *re-fleeting*. Re-timing focuses on changing the time of departure of a flight, in order to create new connections. By adding these new connections, we can increase the capacity of a particular market. Re-fleeting consists of swapping aircraft between flights to better fit demand. To assess our methodology, we generated a series of one-day schedule scenarios with up to 600 flights.

2 Methodology

The Dynamic Airline Capacity ReAssignment (DACRA) algorithm has been designed and conceived as a metaheuristic that combines small changes on the flight departure times (re-timing) and performs swaps to the initial allocation of fleets (re-fleeting). The DACRA algorithm generates pseudo-optimal solutions that allow maximising the occupancy of flights, hence increasing the airline's revenue. The algorithm executes changes on the flight schedule and updates the best solution found so far whenever it finds a new solution that improves the objective function value. To diversify the search, we introduce pseudo-random elements in the market selection process. This way, the algorithm provides a better exploration as it generates a greater number of alternative solutions.

Algorithm 1 DACRA Algorithm

```
 1: bestSol ← GENERATEINITIALSOL(inputs)
 2: while termination criterion is not met do
 3:     newSol ← bestSol
 4:     spillMarket ← SELECTSPILLMARKET(newSol)
 5:     spoilMarket ← SELECTSPOILMARKET(newSol)
 6:     EXECUTERETIMING(spillMarket)
 7:     EXECUTEREFLEETING(spillMarket,spoilMarket)
 8:     if executeRetiming is true or executeRefleeting is true then
 9:         REGENERATEMARKETS(newSol)
10:         REASSIGNPASSENGERS(newSol)
11:         RECALCFLIGHTBUFFERS(newSol)
12:         RECALCOBJFUNCTION(newSol)
13:         if profit(newSol) > profit(bestSol) then
14:             bestSol ← newSol
15:         end if
16:     end if
17: end while
```

Algorithm 1 provides an overview of the DACRA algorithm. As a first step, the current schedule is adapted as the initial solution for the algorithm, and the value of the objective function is calculated. As this is currently the best solution, it is initially stored in the variable *bestSol* (step 1). A copy of the best solution (*bestSol*) is created (*newSol*), which is where changes are made (step 3). In order to generate moves that improve the value of the objective function, we have to select possible markets which allow the application of re-timing and fleet swapping. Two functions are created to select the markets to apply those changes: *selectSpillMarket* (step 4), and selectSpoilMarket (step 5).

The method *selectSpillMarket* takes all markets in the *newSol* solution with a demand greater than the number of available seats and orders them in descending order; i.e markets with a greater gap between demand and capacity (*spill*) will be at the top of the list. Then, using a geometric distribution, the algorithm returns an index position on the list where potential changes will be evaluated. The geometric distribution assigns decreasing probabilities of being selected to the elements of the list. This way, markets with greater spill are prioritised in the search, but it also allows diversifying the search by exploring markets with less spill. Similarly to this procedure, the method *selectSpoilMarket* orders the list in decreasing order of *spoil*, i.e. markets with a higher number of empty seats are at the top of the list. Again, we apply a geometric distribution to choose which markets will be evaluated. After *spillMarket* and *spoilMarket* are selected, the re-timing and re-fleeting processes begin (steps 6 and 7). We only evaluate the possibility of creating new connections for markets with higher demand than the assigned capacity, i.e. re-timing is only applied to spill markets. On the contrary, re-fleeting may benefit both spill and spoil markets. If potential re-timings or fleet swaps are identified, markets are updated in *newSol*, passengers and buffers are reassigned, and the objective function is re-evaluated (steps 9–12). Only if the value of the objective function improves the best solution, the algorithm updates *bestSol*.

3 Results

The DACRA algorithm has been implemented using the Java programming language. All tests have been run in a Toshiba Intel Core i5 with 8Gb RAM. To the best of our knowledge, this problem does not have any benchmark instances, so it highly depends on the information given by airlines. For a first assessment of our algorithm, we use the instance size presented in [1] as a base for our scenarios. All 15 instances correspond to a one-day schedule with up to 600 flights.

#	Forecast Demand	Sold PAX Initial Sol	LF Initial Sol	Sold Pax Best Sol	LF Best Sol	Var. LF (PAX)	Re-timed	Re-fleeted	CPU(min)
1	64612	60471	93.59%	60801	94.10%	330	8	64	7.18
2	62878	58884	93.64%	59090	93.97%	206	9	40	7.37
3	68218	65712	96.32%	65995	96.74%	283	13	100	10.96
4	66092	63726	96.42%	63979	96.80%	253	11	42	10.43
5	66472	63695	95.82%	63866	96.07%	171	16	82	10.68
6	65624	62964	95.94%	63188	96.28%	224	13	52	10.63
7	61265	58976	96.26%	59361	96.89%	385	24	108	10.66
8	65229	62977	96.54%	63099	96.73%	122	9	36	5.37
9	68624	66302	96.61%	66371	96.71%	69	2	44	8.63
10	61509	59826	97.26%	59857	97.31%	31	0	28	8.80
11	65953	60244	91.34%	60564	91.82%	320	0	68	7.45
12	66549	63510	95.43%	63915	96.04%	405	0	108	20.62
13	65727	61691	93.85%	62262	94.72%	571	0	140	20.68
14	66228	60264	90.99%	60889	91.93%	625	0	120	8.85
15	63232	60336	95.42%	60472	95.63%	136	0	84	6.91

Table 1: Results for the 15 instances of the problem

Table 1 shows that after applying the algorithm we have managed to attract an average of 275.4 additional passengers per day (*Var. LF (PAX)*). This represents an increase of almost 1% in the overall occupancy, changing from an initial 94.9% to a 95.6%. On average, 7 flights have been re-timed, while re-fleeting affects an average of 74 flights per day. Thus, 8% of the modifications are attributed to changes in the flight schedule (re-timing) and the remaining 92% consists of changes in the fleet allocation (re-fleeting). Customer satisfaction is generally affected by flight re-timings. Therefore, these results are qualitatively promising, as the majority of changes on the schedule are based on re-fleeting and do not affect the customer directly.

4 Conclusions

Considering the current competitive context in the air transport industry, there is a growing trend focusing on the scheduling aspect and enhancing overall airline processes. In this context, the dynamic flight schedule problem is shown to be a promising area of study.

In this work, we have introduced a new metaheuristic that, by using re-timing and re-fleeting, improves the airline schedule when more reliable demand forecasts are available. Computational results show that, after applying our algorithm, the number of attracted passengers in each one of the instances is increased by an average of 275.4 passengers. Furthermore, we do so in reasonable computational times. Most changes are due to re-fleeting instead of re-timing, minimising the impact on customer satisfaction. Current and future work is focused on adapting the algorithm to balance passenger and airline perspectives, allowing more re-timings. Applying our algorithm to a real airline schedule is also a future development of our research.

References

[1] H. Jiang. Dynamic Airline Scheduling and Robust Airline Schedule De-Peaking. *PhD Thesis*, 2006.

[2] H. Jiang and C. Barnhart. Dynamic Airline Scheduling. *Transportation Science*, 43:336–354, 2009.