META-HEURISTICS GRASP AND ILS APPLIED TO THE AIRCRAFT ROTATION PROBLEM

Alexander de Almeida Pinto

LAVID
Federal University of Paraiba
Joao Pessoa, Brazil

Daniel Goncalvez Ramos

LAVID
Federal University of Paraiba
Joao Pessoa, Brazil

Lucidio dos Anjos Formiga Cabral

IT Department
Federal University of Paraiba
Joao Pessoa, Brazil

Abstract

The Aircraft Rotation Problem refers to the designation of each airplane available at airlines companies such that all planned flights are covered. This is a light constrained problem since any aircraft can be assigned to an unlimited quantity of duty work. However, there are many lower cost solutions that can be obtained by simply adjusting the time of departure of a flight, or even positioning some aircraft by executing ferry flights between two near airports. We describe an algorithm based on GRASP meta-heuristic using ILS as local search, which calculates a sub-optimal way to organize the flights. This is a good solution to large scaled problems, considering that in this case, an exact method would require some years to be applied. Some computational results are presented for a real world problem at Rio-Sul Brazilian Airline.

Keywords: Aircraft Routing, Optimization, Meta-heuristic, GRASP

1 Introduction

The airline industry has been a rich source of problems related to operational research, mainly due to combinatorial explosive nature of these problems. Typically, planning problems involve the creation of lines of work for both aircraft and crew.

On the literature, there are two well-known problems associated to aircraft planning, the Fleet Assignment and the Aircraft Rotation Problem (ARP). The first deals with the allocation of the fleet, determining the type of aircraft used in each flight [1]. The second is the one covered on this paper, and will be explained on the next section.

Due to the combinatorial explosion of the problems involving airlines, they should be solved separately, though each problem should be solved giving conditions to the next one be solved.

Many techniques have been used to give a solution to the ARP, some of them using exact methods, which always obtain the best result, but uses lots of CPU, and in practice is unfeasible, due to the size of real problems. Other techniques try to find a good solution, instead of the best, and try to do that in a fast way. To do so, they use meta-heuristics, a heuristic method that uses some historical knowledge of the solutions combined with some characteristic of the problem to find a good solution for big instances.

1.1 The ARP problem

After solving the Fleet Assignment problem, the one that chooses the kind of airplane that will cover the flights, the problem becomes the organization of the flights. It means, how flights will be connected. The ways to group them can have a very big impact on the cost of the flight network. This cost comes from several factors, like the numbers of aircrafts needed to cover all the flights, the delay between them, the downtime of the flights, and others.

To solve this problem, we must be aware of some constraints involving time and space. For example, an airplane cannot leave before the arrival of the flight that precedes it, neither from a different city that it has arrived on the last flight. Though these are obvious constraints, they make the problem really hard to solve in a large scale, as we could have a very big number of combinations between the connections of the flights. There is also another constraint, related to the time that the airplane has to be on the ground before it leaves. This restriction is usually related to an airport.

Considering these restrictions, we should arrange the flights looking for a cheap network. Its known that finding the cheapest is almost impossible for big instances, but finding a good one can be achieved with some meta-heuristics.

Another important aspect is the maintaining restrictions. It is known that a plane must have periodic checks. Opportunities to perform these tasks occur only on some connections that are potentially available. As consequence, a sequence of flights must be constructed so that these restrictions are not violated. In order to easily incorporate these constraints in our framework, we assume that the rotations are not assigned to specific types of aircraft. Thus, if an aircraft needs maintenance, is created a special flight depart and arrive from the maintenance base chosen. The time of flight is exactly the service time [1].

2 Methodology

In order to find a good solution to the ARP problem, two meta-heuristics were implemented using java Language. On this section, it will be exposed how to represent a solution, as well as some details about the application of the meta-heuristics on the ARP problem.

2.1 Problem Concepts

Each solution has the flight network, with all the connections between the flights. To represent the flights covered by a unique airplane, the concept of a track was applied. So, the goal is to find the smallest number of tracks needed to cover all flights.

Each track is composed by any number of flights, considering the restrictions described on last section. The connection between flights is called arc. An arc has to be one of the followings:

- Simple Connection Arc (type 1): This arc connects flights that dont need to be changed. There is no need to create another flight to connect them too. The cost of this arc is zero.

- Connection With Delay Arc (type 2): With this arc, the second flight needs to be delayed, in order to conform the time restriction. The cost of this arc is the delay (in minutes) of the second flight.
- Connection With Repositioning Arc (type 3): This arc creates a third flight between the two previously selected, in order to conform the space condition. The third flight goes from the destination of the first flight to the origin of the second flight. The cost of this arc is the duration of the created flight.
- Connection With Repositioning and Delay Arc (type 4): Besides creating a new flight to connect the two ones, this type of arc also delays the second flight, so it obeys both space and time restrictions. The cost of this arc is the addition of the delayed time plus the duration of the created flight.

2.2 Problem Solution and Representation

The generation of a solution was divided into two steps. The first one constructs a good solution and the second tries to improve the given solution with some local search algorithms.

To generate a solution, the algorithm GRASP (Greedy Randomized Adaptive Search Procedure) was applied. This algorithm utilizes a greedy heuristic, mixed with a nondeterministic choice of the next element to the solution [FONTE]. To do so, each time an element is to be added to the solution, it is chosen randomically between a fraction of the best ones, according to a greedy algorithm. The algorithm runs several times, and only the best solution goes to the next step.

The second meta-heuristic applied to the ARP is the Iterated Local Search (ILS). It takes a given solution, found on the first phase, and makes some search on different neiborghoods. With this, local minimums can be avoided [Fonte].

It was developed a visual interface to show the solution found by the algorithms. It can be viewed in the results section, picture 1.

3 The constructive algorithm

In this phase, the computer generates a good solution, following the steps:

3.1 Track formation

3.1.1 Initializing a track

To initialize a track, a flight is randomically chosen between the N flights that have the lower departure time and that have not been chosen by another track.

3.1.2 Composing and finalizing a track

After choosing the first flight, or with the previous flight, it is made the choice of the next arch k, $K \in \{1, 2, 3, 4\}$. This choice is made with a probability associated with each arch type P1, P2, P3, P4. The next flight should obey the arch rule and should not have a departure time very far from the arrival of the last. With all the candidates, only the a% best are chosen to be a part of the possible set. So it is randomically picked one and added to the track.

The value of a was defined as 15%, to type 1 arches, and variable, according to the number of candidates, to the other arches.

If the generated set dont have any flight, then the arch type will be changed. If no arch have any possible flight, then the last flight ends the track.

3.2 Composing a network

It is generated as much tracks as all the flights are covered. The constructive algorithm stops when all the flights are already corered.

4 The local search Algorithm

After selecting the best solution generated in construction, it is used some algorithms to try to find a way to improve the solution. It was used the swap and compaction methods and the ILS (Iterated Local Search) meta-heuristic.

There were three kinds of swaps implemented, the Swap-1, Swap-2 and the Swap 3. The swap-n procedure is done by trying to change n connected flights with 1 to n + 1 flights. The change is accepted if there is a cost reduction. The compaction method tries to merge two different tracks into one.

The ILS algorithm was implemented in a way of disturbing the current solution, and then applying the methods described before. Forcing four Swaps-2, not considering if they improve the solution, makes this disturbance.

4.1 Computational results

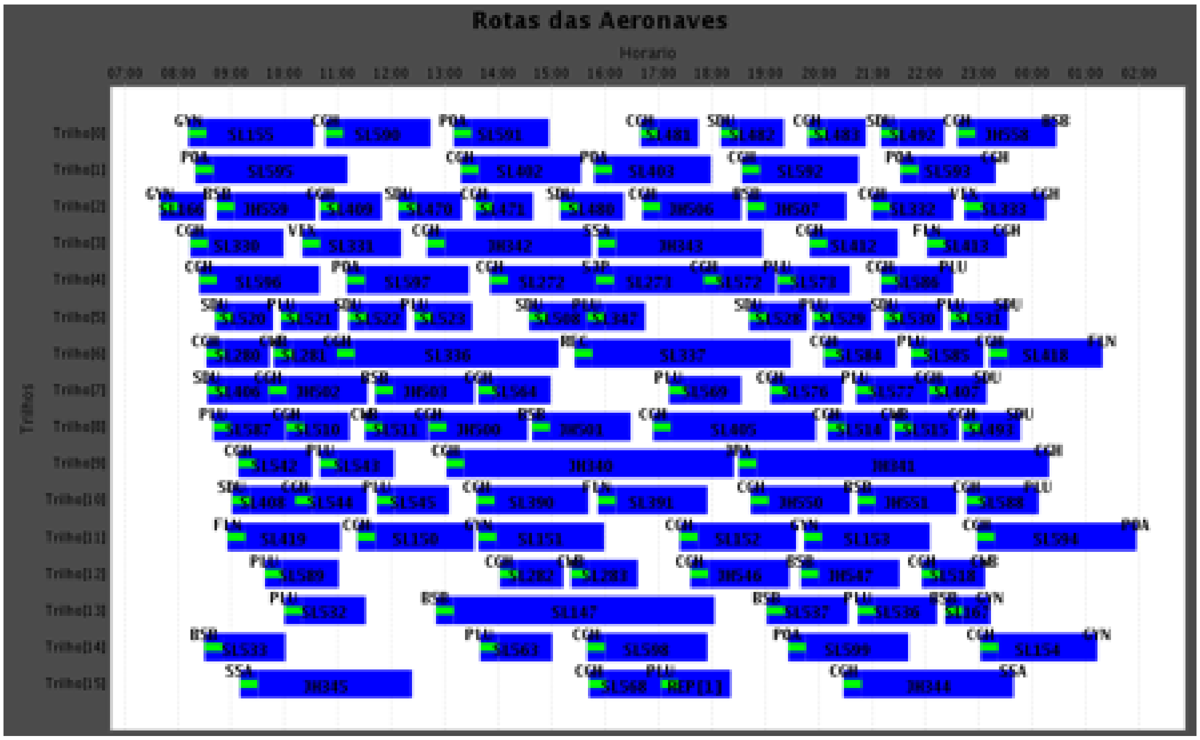
To run the tests, it was used a small daily flight network from Rio-Sul airlines [1]. All data is described on Appendix A to D. With this information, any program can be tested with the same entrance, verifying its efficiency.

All tests run on a Intel Core 2 T5300 1.73GHz, with 1GB DDR-2 RAM. The algorithm was implemented using Java, on NetBeans IDE 6.0.1.

The table below shows details of the achievement of the best solution:

Number of tracks	Cost	Time(seconds)
21	21335	0.039
17	17100	0.113
17	17078	0.315
17	17074	0.346
17	17066	6.012
17	17054	9.612
17	17048	14.179
17	17040	15.0
17	17033	19.484

Below we have a visualization of the best result obtained:



This result can be viewed in more detail in Attachment C.

In the test described was set a ground time standard for all airports, which is 20 minutes. The time between two airports can be found in Attachment B

4.2 Conclusion and Future Researches

The aircrafts are one of the most expensive resources on aeronautics, so every flight planning should be done for a cost reduction. The ARP is one of the problems that can lead to a very high operational cost to a company, if erroneously treated or ignored.

With simple time changing or creating a repositioning flight, the flight network could have a substantial improvement, decreasing aircrafts downtime and reducing the number of planes needed to cover all the defined flights.

For next researches, some efforts are being made to idealize a more complete model of the ARP, leading with weekly tasks, and also solve the next problem to the ARP, the Crew Scheduling. On this next approach, not only the flights will be organized, but also all the crew will be designated to cover the network.

References

[1] Pimentel, A. G., *Uma abordagem heurística para a solução de problemas de recobrimento de conjuntos de grande porte, com aplicação à alocação de tripulações para companhias aéreas*, Tese de Doutorado, Universidade Federal do Rio de Janeiro (2002).

[2] Pontes, R., Lucena, A. e Cabral, L., *Exact and Heuristic Techniques for Solving the Aircraft Rotation Problem*, Technical report Department of Statistics UFPB (2002).

[3] Rezende, M, *Greedy Randomized Adaptative Search Procedures (GRASP)* , Technical Report, ATT Labs Research (1998).

[4] Rubin, J. , "A Technique for the Solution of Massive Set Covering Problems with Application to Airline Crew Scheduling," Transportation Science (1973).

[5] Clarke L, Johnson E, Nemhauser G, Zhu ZX, "The Aircraft Rotation Problem," Annals of Operations Research (1997) 69: 33-46

[6] Klabjan, D. Johnson, E. Nemhauser, G., "Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching," Technical Report, Georgia Institute of Technology (1999).