Application of the Simulated Annealing with Adaptive Local Neighborhood Search to the Tail Assignment Problem

The Case Study of TAP

Francisco José da Silva Montoito

Department of Engineering and Management, Instituto Superior Técnico, 2016

Abstract

The airline industry is present in a strongly competitive market. In recent years, low-cost carriers have entered the market with new business models, forcing traditional airlines to decrease flights price. In an effort to maintain the profit margins, airlines are now seeking for strategies to improve their operational efficiency and therefore reduce the overall operating costs.

The main Portuguese carrier TAP is currently looking for opportunities to improve their current operations. TAP has identified their planning process, specifically the tail assignment phase, as one of their top priorities to improve efficiency. The objective in tail assignment is to define which aircraft should operate which flight.

In this paper, we analyze potential limitations and present a simulated annealing algorithm with adaptive neighborhood search that minimizes the operational costs, while considering at the demand for each flight. Moreover, we analyze different scenarios that include the limitation of the utilization of each aircraft for a given schedule. Finally, it was also created algorithm to generate an initial feasible solution in a short period of time.

Keywords: tail assigment, fleet assigment, airline operating costs, simulated annealing, adaptative neighborhood search, meta-heuristics

1. Introduction

The airline industry sector is one of the most competitive and dynamic markets in the world. Every year airline carriers appear with new offers, reduced prices and different market strategies. Furthermore, the recent growth of low-cost carriers has contributed to the increase of competition in this sector.

Due to their different value proposition, low-cost airlines can offer cheaper flights when compared with traditional flag carriers like TAP (the main Portuguese air carrier). All these factors contributed to the reduction of profit margins across this industry sector (IATA 2014).

The increasing competition in the airline industry market has put more pressure on the management level to persistently reduce costs and increase revenues (Abdelghany and Abdelghany 2012). With market competition becoming stronger the need for solutions that enables cost reduction increases, leading to the implementation of optimization models (Shao 2013). Moreover, the operation research community has had a great influence on the operations of the present air transportation. Motivated by the highly dynamic environment in combination with the complex airline planning system, researchers are now using advanced optimization methods to improve decision support systems and improving the overall airline operations (Yu 2012).

Barnhart et al. (2003) divides the contributions of operations research to the airline industry in three main themes:

1. Revenue management, or yield management, is related with the creation and management of service packages in order to maximize sales;
2. Aviation Infrastructure comprises the design and operation of airports, including the runways, taxiways, aircraft stands and passenger buildings, as also the air traffic flow management.
3. Aircraft and scheduling planning, or more commonly named airline planning process, is traditionally considered as a well-defined succession of activities that depend on established information flows.

Furthermore, Gopalan and Talluri (1998) divided the aircraft and schedule planning into four different phases: schedule generation, fleet assignment, aircraft scheduling, crew scheduling and Disruption Management.

In the aircraft scheduling phase or tail assignment problem, the previously scheduled flights are allocated to individual aircrafts (or tails). This stage can be seen as a resource allocation scheduling problem, where specific resources are allocated to scheduled activities (Kilborn 2000).

Differently from other planning stages, the tail assignment problem has not received much attention by the operation research community. Published studies are limited and fairly recent (Ruther et al. 2013, Başdere and Bilge 2014).

The majority of published studies in this area do not consider the heterogeneity between aircrafts. The main concern in these studies is to find a feasible solution that respects the operational restrictions. Therefore, the individual specifications of each aircraft are seldom used in this planning phase (Gabteni and Grönkvist 2009).

In this paper we propose an efficient tail assignment where the individual specifications of each aircraft, such as fuel consumption profile, are considered.

For this, have we have developed and structure a simulated annealing algorithm that will enable to reach an optimized tail assignment solution that considers at the same time operational restrictions and costs.

In study we will consider two different scenarios, the unbalanced aircraft utilization and the balanced aircraft utilization

Moreover, it is necessary to build an algorithm that generates an initial feasible solution that could be used by the simulated annealing as a first solution.

To conclude, all the algorithms developed during this work are applied to the short and medium-haul operations of the Portuguese carrier TAP. The main objective in this study is to minimize the current operational costs, by making an efficient tail assignment that considers both operational constraints and the demand for each flight.

2. Contextualization of the problem

This paper starts with a short description of TAPs planning process, followed by the description of TAP's short and medium-haul route network and fleet where is detailed the number of aircrafts for each variant. Finally, in the end of this section is discussed the limitations and possible improvements to the current tail assignment method at TAP.

2.1. TAP Planning Process

The planning phase at TAP is a crucial part of the operations. In this process, is created a schedule where the crew, aircrafts, maintenance and flight legs are linked in a final master schedule. In figure 1 is described the four main areas of planning at TAP and the corresponding activities in each one. There are three main areas in the planning process: flight schedule design, aircraft scheduling and crew scheduling. These phases are normally solved sequentially, and the result of an upstream planning stage is delivered into the next downstream planning stage.

Furthermore, tail assignment phase happens before the crew paring phase, where each crew member working at TAP is assigned to a flight, and after, the aircraft routing, where the objective is to find

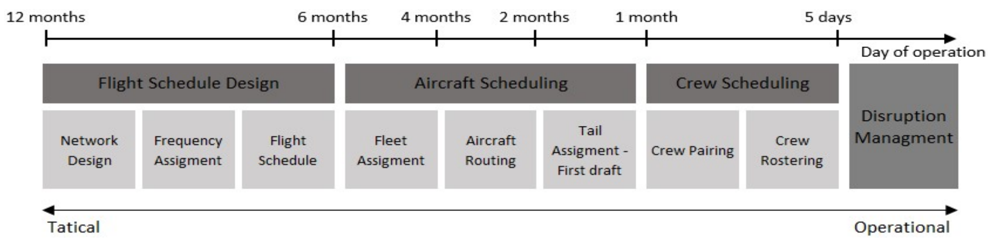

Figure 1 - Time-line of TAP's current planning process at TAP

sequences of flights known in industry as line-of-flights. In all planning phases the only currently objective for TAP is to create a feasible schedule that respects all the network constraints (maintenance services or minimum time required between each flight – turnaround time).

2.2. TAP's Route Network and Fleet for the Short and Medium-Haul Operations

Currently, TAP offers for the short and medium-haul 74 different destinations, corresponding on average 1400 flights per week (for the airbus short and medium-haul fleet), with the regularity varying as a result of the demand behavior. Thus, there is some seasonality in flight sales, where the Christmas and summer months experience higher demands. Additionally, TAP works with a spoke-hub network, where all the flights must depart and return to the same hub (Lisbon or Porto). The spoke-hub model increases the market coverage, as flights are directly or indirectly connected to the main Hub, enabling the maximization on the number of marketable destinations.

In this study we will only focus on the short and medium-haul Airbus fleet. In Table 1 is represented the currently TAP fleet for this operation.

Table 1 - Airbus Fleet for the short and medium-haul operations

Fleet	Manufacture	Variant	Nº Aircrafts
Short and Medium-Haul	Airbus	A319-111 A319-112 A320-214 A321-211	16 5 19 3

Each variant has different seat capacities, engines and the Maximum Takeoff Weight (MTOW). The seat capacity, limits the flights that the aircraft can perform, flights with a higher demand need larger airplanes and flights with a lower demand smaller airplanes. The engines will influence the fuel consumption of each airplane, newer engines will consume less fuel than the older ones. Finally, the MTOW will have an impact on the operating costs of each aircraft, since this value is used to calculate the navigation and landing fees. Furthermore, the fuel costs represent the highest operating cost at TAP, accounting for approximately 50% of the total operating costs.

2.3. Current Tail Assignment Limitations

The existing tail assignment implemented by TAP considers that all the aircrafts are homogeneous. This means that differences between the operational costs of each airplane are not taken into account. The only objective in the current model is to obtain a feasible solution that fulfils all the operational constraints and maintenance activities.

So far the company has not been able to build a model that could represent all the differences between the available aircrafts. For that reason, TAP is not capable to generate accurate information about the costs and possible savings when assigning an aircraft to a specific flight.

To summarize, the current tail assignment is limited and not able to give an optimal solution that minimizes the operational costs. TAP is now seeking for a model that would enable an increase in operation efficiency, since it recognizes that in reality each airplane is unique with distinct characteristics.

3. Solution Approach Methodology

The objective of this work is to build a simulated annealing model that will help TAP allocate the short and medium-haul aircrafts to a large number of flights. The main model is mainly based on the optimization model built by Lapp and Wikenhauser (2012).

3.1. Mathematical model

In the model created we have considered that flights are aggregated whenever is possible. This means that activities that have only a single possible connection are merged, resulting in a single aggregated activity. The starting time and departure of the aggregated activity are given by the first activity, the total duration is given by the sum of the durations of each activity and finally, the arrival time and airport is given by the last activity. The DOCs of the aggregated activities equals to the sum of the costs of each activity.

Also the model considers in the objective function besides the costs previously mentioned, a term in the objective function, that penalizes (increases the cost) whenever a demand for a specific flight surpasses the aircraft seat capacity. The second extension is also a term in the objective function that penalizes the objective function whenever an airplane exceeds the total utilization defined by the user.

Next, we present the mathematical formulation that was used to structure the simulated annealing solution approach.

Index and sets

$a \in A$ – as the set of all the activities
$t \in T$ – as the set of all tails available;
$s, ss \in S$ – as the set of all the airports;
R – as the set of assignment of each activity *a* to the respective inbound (*s*) and outbound airport (*ss*);
IC – as the set of all the activities *a* that cannot be assigned to a given subset of tails *t*;
UP – as the set of all the possible upstream activities for activity *a*;
DN – as the set of all the possible downstream activities for activity *a;*
AD – as the set of all the activities *a* that have a higher demand than the seat capacity of tail t;

Parameters

$fcost$ – jet fuel cost per kilogram;
$putili$ – utilization penalization factor;
$fconsump_t$ – fuel consumption in kilograms per block hour for a given tail t;
$mtow_t$ – maximum takeoff weight in tonnes for a given tail *t*;
$tavail_t$ – initial availability time for a specific tail t;
$seatcap_t$ – total number of seats for a given tail t;
$maxutili_t$ – maximum utilization percentage for a specific tail *t*;
p_t – weight factor for a given tail t.
$urate_{s,ss}$ – unit rate of the airport-pair s, ss;
$dfactor_{s,ss}$ – distance between each airport-pair s, ss;
$lcharge_{ss}$ – landing charge for the destination airport ss;
$tmin_{ss}$ – minimum required turnaround time for each airport-pair s, ss;
$tduration_a$ – total duration time in block hours of each activity a;
$fpercent_a$ – percentage of the flight a block hours in the total block hours of the whole set A;
$Tprice_a$ – average ticket price activity for activity a;
$fdemand_a$ – total number of tickets sold or forecasted for activity a.

Decision Variables

$w_{a,t}$ – is 1 if the activity a is the first activity to be assigned to tail t and 0 otherwise;
$x_{a,t}$ – is 1 if the activity a is assigned to tail t and 0 otherwise;
$y_{a,t}$ – is 1 if the activity a is the last activity to be assigned to tail t and 0 otherwise;

Mathematical Formulation

$$\begin{aligned} &\boldsymbol{min} \sum_{a \epsilon A:R} \sum_{t \epsilon T} \left[fcost \times fconsump_t \times tduration_a \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] + \\ &\sum_{a \epsilon A:R} \sum_{s,ss \in S} \sum_{t \epsilon T} \left[p_t \times urate_{s,ss} \times dfactor_{s,ss} \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] + \\ &\sum_{a \epsilon A:R} \sum_{ss \in S} \sum_{t \epsilon T} \left[mtow_t \times lcharge_{ss} \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] + \\ &\sum_{a \epsilon A:R} \sum_{t \epsilon T} \left[mcost_t \times tduration_a \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] + \\ &\sum_{a \epsilon A:R:AC} \sum_{t \epsilon T} \left[(fdemand_a - seatcap_t) \times tprice_a \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] + \\ &\sum_{a \epsilon A:R:AC} \sum_{t \epsilon T} \left[putili \times (fpercent - maxutili_t) \times \left(\sum_{a \in A:DN} x_{a,t} + w_{a,t} + y_{a,t}\right)\right] \end{aligned} \quad (1)$$

Subject to:

$$\sum_{a,aa \epsilon A} w_{a,t} \leq 1, \forall t \in T \quad (2)$$

$$\sum_{a \in A:DN} \left(x_{a,aa,t} + y_{a,t}\right) = 1, \forall t \in T \quad (3)$$

$$\sum_{a,aa \in A:UP} x_{a,aa,t} + w_{a,t} = \sum_{a,aa \in A:DN} x_{a,aa,t} + y_{a,t}, \forall t \in T \quad (4)$$

$$w_{a,t} \times tavail_t \leq tavail_t, \forall t \in T \wedge a \in A \quad (5)$$

$$w_{a,t}, x_{a,aa,t}, y_{a,t} \in \{0,1\}, \forall t \in T \wedge a \in A \quad (6)$$

Objective function (1): In the first term is calculated the total fuel consumption of the entire fleet, which is dependent of the current jet fuel price, the consumption profile of each tail and the duration of each flight. The second term is related with the navigation costs that are dependent on the distance between the departure and arrival airport, the MTOW of the airplane and the specific rate for each connection. The third term gives the total cost of the landing charges. This value is calculated based on the MTOW of the airplane and the landing charge at the arrival airport. In the fourth term is calculated the total maintenance cost that depends on the average maintenance cost per block hour and the number of block hours per flight. The fifth term is a penalization factor that increases the cost of the solution, whenever the demand for a specific flight is higher than the seat capacity of the aircraft. The sixth and final term is also a penalization factor, but related with the utilization of a specific tail.

Constraints: Constraint (2) ensures that each aircraft has no more than one first activity assigned to it $w_{a,t}$ is equal to zero, it means that the aircraft is not used in on the schedule being study). Constraint (3) guarantees that every activity as one and only one aircraft allocated to that activity. The network flow balance constraint (4) forces each aircraft to follow a feasible sequence of activities. Constraint (5) ensures that a specific tail is only assigned to an activity after becoming available at the beginning of the schedule (some tails might have assignments resulting from a previous schedule or being unavailable due to maintenance). Finally, constraint (6) defines the decision variables domain.

3.2. Simulated Annealing Algorithm

The simulated annealing process was three distinct phases in order to find the best solution for a given schedule: Initialization, Iterative process and Stopping. In the initialization process is defined all the simulated annealing parameters and is created an initial solution. In the iterative process are created new solutions that are evaluated and that can be accepted or not. In the last phase, if the stopping criteria are meet than the model stops and returns the last reached solution. In Figure 2 explained the algorithm procedures used for this work.

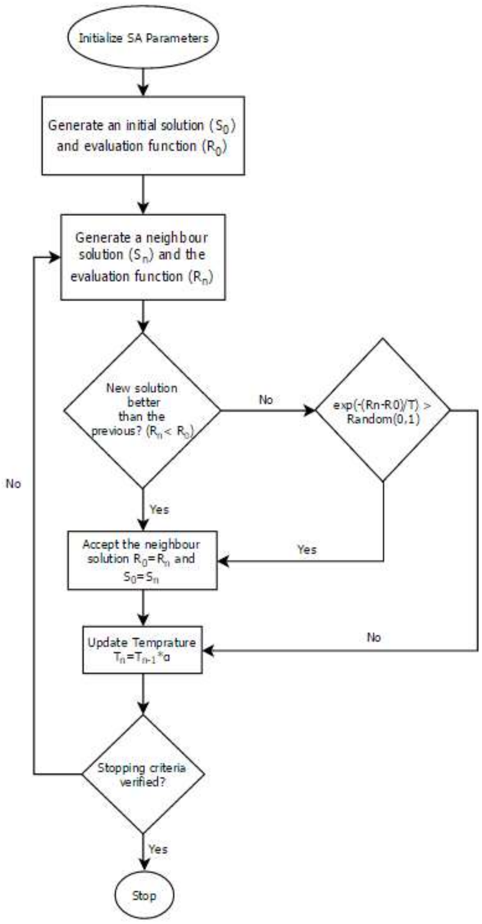

Figure 2 - Simulated annealing algorithm flow diagram

1. The first procedure is definition of the initial parameters that will be used during the algorithm. These include, the initial temperature, the annealing schedule (number of iterations at the same temperature) and the stopping criteria (minimum temperature);

2. In this step the algorithm generates an initial feasible solution, using the model described in Section 3.3, and uses the objective function to evaluate the initial feasible solution;

3. Here is defined a new solution, based on an adaptive local neighborhood search algorithm that is explained further in Section 3.2.1, and it is evaluated based on the objective function;

4. The new solution is compared with the previous accepted solution. If the new solution was a lower cost than the previous, this solution is automatically accepted. When the neighbor solution is worse (higher cost), the probability of acceptance is defined by equation (7):

$$P_{acceptance} = e^{-\frac{(NewSolution - PreviousS\qquad)}{Temprature}} \quad (7)$$

To accept the new solution a random number between [0, 1] is generated. If $P_{acceptance} > Randomnumber$ the new solution is accepted, otherwise the neighbor solution is rejected.

5. After a defined number of iterations the temperature decreases by a cooling constant α, where temperature is defined by the formula: $T_n = T_{n-1} \times \alpha$ (8). This enables the solution to escape the potential local optimum in the beginning of the algorithm (higher temperatures), and to fix the solution at the end of the algorithm (lower temperatures).

6. The model stops after n consecutive temperature levels, wherein the difference between the actual solution and the best solution found during the process is equal or inferior to 0.01% of the value of the best solution, $(R_0 - R_{best}) \leq 0.0001 \times R_{best}$ (9). This criterion keeps the algorithm running until the probability of accepting a new solution is very low. If the stopping criteria are met the algorithm stops, returning the current solution, if not, is initiated a new iteration.

All the parameters including the initial temperature, the number of iterations before temperature updating and the cooling constant (α) are analyzed and defined in Section 4.

3.2.1 Adaptive Neighborhood Local Search

The adaptive search enables the use of different search methods within the same optimization algorithm. Each method is chosen randomly, where the probability of being selected depends on the performance of previous iterations.

The probabilities are calculated at the same time of the simulated annealing temperature readjustment. These probabilities are defined by calculating the total amount that a specific method contributed to the objective function divided by the time spent on that method (8). These ratios are then compared with each other and the higher is the ratio, the higher is the probability of method to be chosen in future iterations.

$$r_{improvement}\,(n) = \frac{Total\ improvementof\ method\ n}{Total\ time\ spent\ on\ method\ n} \quad (10)$$

After the ratio comparison between all search methods, the probability is defined based on predefined values. These are further defined in Section 4.

For this work we have created two different local search methods: activity change and Line-of-activities change.

Activity Change Method

This method randomly searches an activity and tries to change also randomly the aircraft that is currently assigned. If the selected activity has only one aircraft available to perform that activity, the model continuous the search until is found one activity with two or more available aircrafts.

Line-of-Activities Change Method

This method is consists in swapping the line-of-activities of two aircrafts at a given point in schedule. The method starts by searching randomly for an activity with at least one available aircraft, besides the currently assigned aircraft, to perform the activity. The aircrafts selected to swap the line-of-activities must not have a maintenance activity after the selected activity, otherwise the method starts the search again. When these two conditions are met, the line-of-activities from the selected activity until the end of the planning horizon are swapped between the two aircrafts.

3.3. Initial Feasible Solution Algorithm

In order to generate an initial feasible solution, a First-in, First-out (FIFO) algorithm was created. The method starts in the first activity of the schedule, and assigns the aircrafts based on the method chosen. The assigning method can be: greedy (the aircraft with the lowest cost is assigned), partially greedy (the aircraft is chosen randomly between the best 5 cost efficient aircrafts) and random (the aircraft is chosen randomly).

In a cycle, the algorithm goes through every single activity, assigning an airplane to each one. This process is repeated until the final activity is reached. When the algorithm encounters a pre-assigned maintenance activity, the activity is skipped without assigning an aircraft.

4. Determination of the Simulated Annealing Initial Parameters

It is extremely important to define well the initial parameters for the simulated annealing algorithm, as they have a great influence on the overall performance of the algorithm. We start by analyzing the initial temperature, then the stopping criterion and the annealing schedule, and finally the search method selection probability.

Initial Temperature

In the created simulated annealing algorithm we have defined two distinct neighborhood search methods with different output magnitude values. This means that if we want define the same probability of acceptance for both methods, we have to use two distinct temperatures.

To define the initial temperature *T* for both local search methods, we have made a total of 10,000 iterations for each method separately, where we listed the all the positive ΔE $(New\ solution - Old\ solution)$ values. Using the formula $T = \Delta E / \ln(p)$ (9) we have calculated for each ΔE the initial temperature needed for an acceptance probability of *p* = 0.8. Finally, we made an average of all temperatures calculated. For the activity change local search method we have calculated an initial temperature of 3,700 and for the line-of activities method we have defined an initial temperature of 5,500.

Stopping criteria

To define the number of consecutive temperature levels *n* before stopping the algorithm, we have made a test where we defined stopping criteria as 50,000 iterations. Then, for each consecutive level *n* reached (e.g. 1, 2, 3, *n*+1) the algorithm registered the number of the iteration. In Figure 3, is shown the evolution of the objective function and the iteration number where n consecutive levels (5, 10, 15, 20, 25 and 30) where reached. By analyzing the graph we can see that when n is equal to 20 (iteration 37,899) the objective function solution starts to stabilize. After this point, the solution values experience much less variance than the values before. Although, the 20 consecutive levels provide a good stopping point, we have decided to use in this work 23 levels as a “safety margin”.

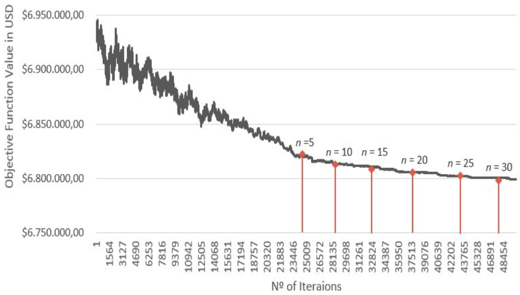

Figure 3 - Objective function evolution and number of consecutive n levels

Annealing Schedule

To define the α factor we have tested the solution approach for 4 different values of α: 0.9, 0.95, 0.975 and 0.99. In Table 2, is represented for the four test instances, the final objective function value, the number of iterations made the solving time, and the comparison between the results.

Table 2 - Algorithm results for α equal to 0.90, 0.95, 0.975 and 0.99

α	Objective Function Value in USD	Solving Time (s)	Comparison with the Best Solutions	
			Obj. Func.	Solving Time
0.90	6,839,295	305,6	+ 0,5%	-
0.95	6,824,625	593,25	+0,3%	+ 94,4%
0.975	6,808,400	912.8	+0,1%	+ 199%
0.99	6,803,383	1,876.6	-	+ 536,9%

For our solution approach, we have decided to define α as 0.975. This cooling factor represents the better tradeoff between the solving time and the objective function value. The value for the objective function is only 0.1% worse then the best solution, and the total solving time is 15 minutes shorter.

Local Search Method Selection Probability

For this work we have tested three different levels of probabilities. Additionally, the probabilities are adjusted at the same time as the temperature levels, 200 iterations. In Table 3 , is represented the three levels of probabilities with the respective objective function value, solving time and the number of iterations made for each search method. By analyzing the table, we can see that all of the probability levels have about the same solving time, being the level 85%-15% the one with the lowest solving time. The best value for the objective function is obtained when using the 75%-25% level.

Additionally, we can see that the activity change local search method is selected more times for this schedule. For this work we have selected the 75%-25% percentage, due to the smallest objective function cost.

Table 3 - Probability levels to select the local search method

Probability of the Local Search Method to be Chosen		Objective Function Value in USD
Best Past Performance	Worse Past Performance	
65%	**35%**	6,832,069
75%	**25%**	6,813,625
85%	**15%**	6,820,623

5. Results

For the results analysis in this study four schedules are considered, one for the high season period and the other for the low season period. In Table 4, are represented the four schedules.

5.1. Initial Feasible Solution (FIFO)

To assess the different assigning methods for the FIFO algorithm, we have used the HS7 schedule, as it represents the usual scheduling period used by TAP in the most demanding period (higher number of activities). In Table 5, is shown the total cost, the solving time and the comparison with the original solution.

The FIFO algorithm with the greedy assignment method returns the best solution in the shorter amount of time, when compared with the other algorithms. The solution for this algorithm costs less $9,045, than the partially greedy assigning method. Furthermore, the FIFO greedy solution shows lower costs for 4 out of the total 5 components of the operational costs. The largest difference is in the fuel costs where the original solution is $28,127 more expensive than the FIFO approach. On the other hand, the FIFO solution does not meet the entire demand for this given schedule, resulting in a penalization for the final solution.

Table 6 - FIFO algorithm results for HS7

	Total Cost (USD)	Comparison With the Original Solution	Solving Time (s)
Greedy	6,881,295	$ -39,989	10,4
Partially Greedy	6,901,352	$ -19,932	12,6
Random	6,939,340	$ 18,056	11,5

In Figure 18, is represented the comparison of the utilization in BHs by variant for the original and FIFO greedy solutions. As shown in the chart, there is decrease in the utilization of the A321 and A320 variants and subsequently an increase in the utilization of the variant A319. This transition in the utilization of the variants, is mainly related with the total fuel cost.

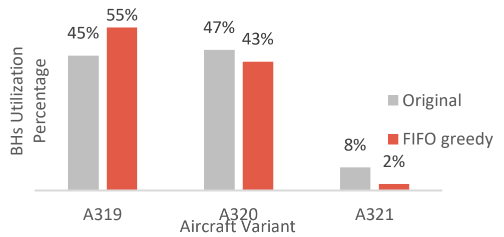

Table 4 - Comparison of the utilization percentage for each variant, between the Original Schedule and FIFO greedy solution schedule

To conclude, the FIFO greedy algorithm provides a good solution with approximately 40 thousand dollars in savings when compared with the original solution.

Furthermore, the solving time for this algorithm is very short, making it a good a first comparison with the original solution, or

Table 5 - Scenarios considered for this study

Season Period	Instance	Schedule Period		Nº of Activities		Nº of Available Aircrafts	Block Hours	Original Solution Cost
		Start Time	Start Time	Single	Aggregated			
Low Season	LS5	01/03/16	05/03/16	788	395	41	1936	$ 3,935,410
	LS7	06/03/16	12/01/16	1171	592	42	2925	$ 5,874,500
High Season	HS5	01/07/16	05/07/16	1011	489	43	2397	$ 4,840,198
	HS7	01/07/16	07/07/16	1421	699	43	3430	$ 6,921,284

even be used by TAP if there is no time to run the simulated annealing algorithm.

5.2. Simulated Annealing Algorithm

To test the performance of the created simulated annealing algorithm, we used all the schedules previously defined. In Table 7 is represented the comparison between the original solution with the optimal one and the solving time for each one of the tested schedules.

When we compare the two five days schedules (LS5 and HS5), it is noticeable that the savings in the HS5 are higher than in LS5. This happens because there is a higher amount of flights in the high season period, making it possible to create more saving opportunities. On the other hand, the savings percentages are very similar, with a slightly higher percentage savings for the HS5 schedule.

For the 7 days schedule period, like in the 5 days schedule, the high season period has the higher absolute and relative value for the savings, when compared with the low season period.

If we compare the solving times between the 5 day schedules and the 7 days schedules, it is observable that the variance amid the solving times is relatively small. For the high season schedule, the difference between the 5 and 7 days is 417 seconds (approximately 7 minutes), and for the low season the difference is even smaller, only 164.2 seconds (approximately 2 minutes and 30 seconds) between the two solutions.

For all the schedules created, the algorithm was able to find a lower cost solution, that would enable TAP to reduce the direct operational costs.

After this general analysis, we carry a more extensive analysis for the LS7 schedule.

Low Season – 7 Days (41 Aircrafts Available)

For the 7 days low season schedule, it was required 37,001 iterations to reach the optimum total cost of $5,810,681, where 23,082 iterations were made using the line-of-activities change method and the rest 3,918 were made using the activity change method. In Figure 5, is represented chart with the starting point of the solution ($6,014,230) and the progress made until the final solution is reached.

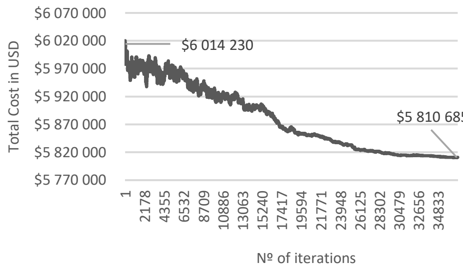

Figure 4 - Evolution of the objective function for the LS7 schedule

In the same way as FIFO algorithm with greedy assignment, the A319 aircrafts are preferred (mainly due to the lower fuel consumption profile) than the other two variants, for an optimum solution. For this schedule the original solution considered an average utilization for the A319 of 58 flight hours per aircraft for the optimized solution this value rises to 69 hours. For the A321 (highest fuel consumption aircraft), in the original solution this variant has 54 flight hours assigned, contrarily for the optimized solution this value decreases to 11 flight hours per aircraft.

Furthermore, the highest savings are obtained by the reduction of the fuel costs, representing a total of $39,037 in savings. The second highest saving is for the

Table 6 - Simulated annealing results for the scenarios considered

Test Instance	Objective Function Value in USD: Simulated Annealing Solution	Objective Function Value in USD: Original Solution	Savings In USD	Savings Percentage	Solving Time (s)
LS5	$ 3,835,448	$ 3,899,627	$ 64,180	1,7%	647,7
HS5	$ 4,750,756	$ 4,840,198	$ 89,442	1,8%	699,5
LS7	$ 5,874,500	$ 5,808,761	$ 65,740	1,1%	811,9
HS7	$ 6,810,761	$ 6,912,284	$ 110,707	1,6%	1033,04

maintenance cost that represents a total amount of $ 19,698.

5.3. Simulated Annealing Algorithm with Balanced Aircraft Utilization

In this scenario we study the impacts to the final solution, when we define a penalization that imposes a balanced aircraft utilization. In order to simplify this analysis, we have considered only the LS7 schedule. Furthermore, for the penalization cost factor we have considered $ 5,000,000 (this cost was achieved by trial and error). This cost is considered by the algorithm, every time that an aircraft surpasses the utilization limit. As an example, if an aircraft that has a defined maximum utilization of 2% of the total flight hours in a schedule, if this value of utilization is exceeded, the aircraft receives a penalization that is directly proportional to the new utilization percentage.

The final savings for this solution approach, were much lower when compared with the original solution. For this solution we only get a total savings of $ 10,055 when compared with the original solution. Moreover, the solving time was quite high, around 71 minutes, due to the low convergence of the solutions.

When we compare the utilization of the aircrafts between all the solutions, as shown in Figure 5, we can see that in the balanced aircraft utilization approach, the usage between all three variants is much more balanced than in the solution approach with unbalanced aircraft utilization.

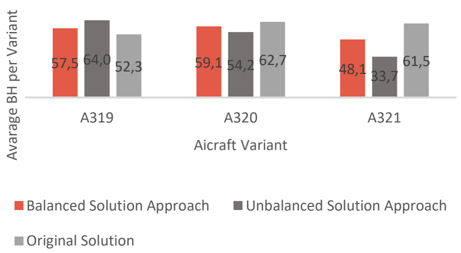

Figure 5 - Average flight hour per Variant

5.4. Sensitivity Analysis

The fuel costs and the maintenance costs were varied in -10%, +20% and -20%, +20% respectively. This analysis proved the algorithm robustness, as the model continued to assign the similar percentage for each variant. When the costs increase the savings opportunity are greater.

6. Conclusions

The developed FIFO algorithm with greedy assignment has shown that TAP would be able to save around $40,000 for a typical high season weak, with a very low solving time (10 seconds). Furthermore, the simulated annealing algorithm has proved that is able to reduce greatly the total direct costs in all schedules studied, by assigning more efficient aircrafts to each flight.

The future directions for this work involve: a robust tail assignment approach, tail assignment as a profitability problem and the distinction between the economic class and business class.

7. References

Abdelghany, K. and A. Abdelghany (2012). Modeling applications in the airline industry, Ashgate Publishing, Ltd.

Barnhart, C., P. Belobaba and A. R. Odoni (2003). "Applications of operations research in the air transport industry."

Başdere, M. and Ü. Bilge (2014). "Operational aircraft maintenance routing problem with remaining time consideration." European Journal of Operational Research 235(1): 315-328.

Gopalan, R. and K. T. Talluri (1998). "Mathematical models in airline schedule planning: A survey." Annals of Operations Research 76: 155-185.

Kilborn, E. (2000). Aircraft scheduling and operation–a constraint programming approach, Chalmers University of Technology and Göteborg University.

IATA (2014). Profitability and the Air Transport Value Chain.

Lapp, M. and F. Wikenhauser (2012). "Incorporating aircraft efficiency measures into the tail assignment problem." Journal of Air Transport Management 19: 25-30.

Ruther, S., N. Boland and F. Engineer (2013). "Integrated aircraft routing, crew pairing, and tail assignment: branch-and-price with many pricing problems."

Shao, S. (2013). Integrated Aircraft Fleeting, Routing, and Crew Pairing Models and Algorithms for the Airline Industry. Doctor of Philosophy, Virginia Polytechnic Institute and State University.

Yu, G. (2012). Operations research in the airline industry, Springer Science & Business Media.