Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 58 (2021) 53–60

Transportation Research Procedia

www.elsevier.com/locate/procedia

14th Conference on Transport Engineering: 6th – 8th July 2021

A Novel Approach to the Tail Assignment Problem in Airline Planning

Manuel Fuentes[a], Luis Cadarso[b,]*, Vikrant Vaze[c], Cynthia Barnhart[d]

[a]*Cisneria Engineering, Alcalá de Henares, Madrid, 28807, Spain*
[b]*Aerospace Systems and Transport Research Group, European Institute for Aviation Training and Accreditation (EIATA), Rey Juan Carlos University, Fuenlabrada, Madrid, 28942, Spain*
[c]*Thayer School of Engineering, Dartmouth College, 14 Engineering Drive, Hanover NH 03755, United States of America*
[d]*Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States of America*

Abstract

Combinatorial optimization problems abound in the field of airline planning. Aircraft and passengers fly on networks made up of flights and airports. To schedule aircraft, assignments of fleet types to flights and of aircraft to routes must be determined. The former is known as the fleet assignment problem while the latter is known as the aircraft routing problem in the literature. Aircraft routing is typically addressed as a feasibility problem, the solution to which is required for the construction of crew schedules. All these issues are typically resolved 4 to 6 months before the day of operations. As a result, there is little information available about each aircraft's operational status when making such decisions. The tail assignment problem is solved when additional information about operational conditions is revealed, with the goal of determining each aircraft's route for the day of operations while accounting for the planned aircraft routes and crew schedules. It is a problem that must be resolved close to the day of operations. We propose a mathematical programming approach that captures all operational constraints and maintenance requirements while minimizing operational costs and schedule changes relative to original plans. The computational experiments are based on realistic cases drawn from a Spanish airline with over 1000 flights and over 100 aircraft.

Keywords: airline; tail assignment; planning; aircraft routing.

* Corresponding author. Tel.: +34-914888775.
E-mail address: luis.cadarso@urjc.es

10.1016/j.trpro.2021.11.008

1. Introduction

Tail assignment is the step in the airline planning process in which specific aircraft (tails) are assigned to flights on a specific schedule. This task is completed a few days before the operation and is subject to several constraints. Firstly, and foremost, all maintenance activities must be ensured. Secondly, information that becomes available following fleet assignment and aircraft routing should be considered. Finally, all operational constraints must be adhered to. The availability of information is the reason for performing tail assignment in the operational horizon. Little to no information about the maintenance needs of each tail is known months or weeks before the day of operations; therefore, generic maintenance opportunities are only considered in the aircraft routing problem. Also, as late adjustments in the schedule occur, the series of flights to be flown by the same tails (line-of-flights) can be modified; certain flights can be cancelled, and others can be added immediately before the service.

The aircraft routing problem has been studied for decades and described in a variety of ways (Barnhart et al., 1998). The following is a generally accepted term. Given the assignment of fleet types to flights, the aircraft routing involves deciding the sequence of flights, i.e., line-of-flights, to be flown by each aircraft and ensuring that each flight is flown exactly once, each aircraft visits maintenance stations at regular intervals, and the solution uses only available aircraft of each type (Desaulniers et al., 1997). It should be noted that it is standard practice to produce flight sequences for aircraft quite early in the planning phase. This early decision is essential to provide input data to the crew planning process and to prepare long-term maintenance, not considering individual constraints but generic maintenance constraints. Then, as the day of operations approaches, the aircraft must be assigned to flight sequences. However, on average, only 80–85 percent of the sequences can be used in actual operations (Liang et al., 2015). This is due to the quality and level of detail of knowledge available about aircraft operational status when solving the aircraft routing, which, on average, is insufficient.

Consequently, the tail assignment problem is addressed very close to the day of operations. Here, all individual operational and maintenance constraints are considered. The problem is solved for a time horizon, which usually spans several days, and provides fully operational assignments of aircraft (tails) to sequences of flights. In practice, the input sequences of flights, which are determined in the aircraft maintenance routing problem, are usually not all suitable for satisfying operational constraints on the day of operations and they must be updated (Liang et al., 2015). Some variants of this problem have been developed in response to different airline business practices and involving planning horizons from months to days (Maher et al., 2018; Grönkvist, 2005; Gabteni and Grönkvist, 2009).

Because of the different approaches to the problem in the literature and real practice, there is not a widely accepted objective function. For example, the aircraft routing problem is not universally recognized as an optimization problem. Many authors have defined it as a feasibility problem, and others have attempted to maximize same-aircraft connections for passengers or to minimize expected delays (Lan et al., 2006). Also, it is common practice for airlines addressing the problem to consider it to be a feasibility problem (Grönkvist, 2005).

A mathematical model, which minimizes constraints violations and conflicts among others, and a solution strategy are proposed to find optimal solutions to real-world large-scale tail assignment problem instances in a short amount of time. The mathematical formulation presented here is based on the one developed for the crew scheduling problem, and it can capture all of the operational requirements for problem instances as given by our airline partners, including apron-related requirements. Existing methods in the literature fail to capture such a level of detail. Attempts to achieve this while still providing good quality solutions have failed. A rolling horizon approach is used to solve large-scale instances and achieve feasible solutions, in which the model is solved in many smaller sub-models. This approach is useful for finding a viable solution to a problem in a limited amount of time. The obtained solution is then fed into a heuristic-based method, which allows us to empirically demonstrate optimal solutions to the overall model. Computational studies based on data from Vueling Airlines, one of Spain's major airlines, are presented. The model's solutions outperform the airline's solutions, according to empirical proof. Besides, as evidence of the presented approach's effectiveness, a decision support tool was developed and given to the airline. This tool, which is based on the methods described in this report, is now being used in real-world operations.

2. Problem description

A time-directed graph represents the airline network, with nodes representing tasks such as flights and maintenance duties, and arcs representing relations between tasks. An origin airport, a departure time, a destination airport, and an arrival time define tasks. One tail must be assigned to each task. The remainder of this section introduces rotations, maintenance duties, the guidelines to be followed when assigning tails, and the goals to be pursued.

Rotations. A rotation is a series of flights performed by the same tail on the same day. A Line of Work (LoW) is a series of rotations to be performed by the same tail over a set period of time, usually several days. It should be noted that rotations and LoWs are results of the aircraft routing problem, which is solved after flight schedules and fleet assignments are determined. Rotations and the fleet allocated to them are the key inputs in tail assignment, and they should be kept as consistent as possible when allocating tails to tasks because crew schedules are dependent on them.

However, if planned rotations become infeasible, for example, due to tail operational conditions, the tail assignment must update them such that the flight schedule remains feasible. When allocating tails to rotations, the maintenance requirements of the tails must be considered, or else assignments will be infeasible. Swaps are used to help or make it easier to meet those requirements. A swap is the recombination of sections of two rotations with different bases, allowing the tails assigned to them to switch bases. Some rules, such as those requiring space-time compatibility, must be followed by swaps. Furthermore, their effect on crews must be minimal, so only a limited collection of all the potential swaps is allowed, as defined when the aircraft routing problem is solved and following the airline's criteria.

Maintenance duties. Maintenance entails arranging the repair of identified issues, replacing objects after a certain amount of flight hours or calendar time, correcting previously found defects (e.g., pilot or crew reports, line inspection, items postponed from previous maintenance), and conducting scheduled maintenance. On an aircraft, various types of maintenance duties must be performed, which may or may not be planned ahead of time. Scheduled duties are identified in advance since they must be completed on a regular basis, while others are the product of operations, typically following equipment failures. They can be classified based on their frequency: flight line inspections are carried out on a regular basis, overnight checks (also known as "daily checks") are small checks conducted every two days during the night, A checks are light checks that are performed every few hundred flying hours, B checks are light checks that are performed every few months, C checks are heavy checks that are performed every 2 years, and D checks are heavy checks that are performed every 6–10 years. The aim of these checks is to perform both routine and non-routine aircraft maintenance. It should be noted that some maintenance checks (hereafter referred to as maintenance tasks) are more flexible than others when conducting tail assignment. Since the tail assignment planning horizon features several calendar days, flight line checks and overnight checks are thoroughly considered. The remaining checks are also fully considered but are well established prior to operations.

Allocation rules. Several rules govern the assignment of tails to tasks. They can be classified as hard or soft requirements. Any hard requisite must always be met, while soft ones may be related to market considerations, such as a preference for particular tails due to capacity or efficiency, and may be violated, but, if violated, have a negative effect on the solution efficiency. The number of current rules is usually enormous, and they are complicated, making this problem difficult for planners to solve. They can be general or global rules, but they can also be fleet, aircraft, or airport specific (Grönkvist, 2005).

The following is an example of a hard constraint. Owing to noise restrictions, a specific tail cannot operate at a specific airport at certain times. And an example of a soft constraint follows. Because of its maximum takeoff weight, an aircraft type has limited performance at a specific airport. It will fly from that airport if the tail is not at its maximum weight.

Objectives. The tail assignment problem can have several objectives. They can change as the revealed information develops as the day of operations approaches. The primary goal in the early stages is viability, while in the later stages, meeting optimization requirements becomes more important. The following are some of the most important key performance indicators to be minimized: soft restriction violations, fleet changes, swap use, prolonged idle periods, fuel costs, and apron conflicts. Simultaneous departures of neighbouring aircraft on the apron or ramps may cause conflicts and delays. They could need the same airport services at the same time. As a result, they should be kept to a minimum.

3. Mathematical model

We propose an Integer Linear Programming (ILP) model. Its mathematical formulation is built on a framework in which the tasks to be assigned are nodes and the relations between them are arcs. Mathematically, we consider one type of node (we treat all tasks the same, regardless of their nature) and one type of arc. The model's employed sets, parameters, and variables are described next. The mathematical formulation is then explained.

3.1. Sets

- F is the set of tasks to be covered in the given time-horizon. Tasks are indexed by i and j.
- $FF \subseteq F$ is the subset of tasks which are flights.
- $FSM \subseteq F$ is the subset of tasks which are soft maintenance tasks. They can be postponed if necessary, to improve schedule performance. If they are postponed, they must be rescheduled.
- $FHM \subseteq F$ is the subset of tasks which are hard maintenance tasks. They cannot be postponed.
- $F_i^+ (F_i^-) \subseteq F$ is the set of tasks which may follow (precede) task i in a line of work.
- P is the set of fleet types.
- $P_i \subseteq P$ is the set of fleet types compatible with task i.
- T is the set of tails.
- $T_i^F \subseteq T$ is the set of tails compatible with task i.
- $T_p^P \subseteq T$ is the set of tails belonging to fleet type p.
- $F_t^T \subseteq F$ is the set of tasks which may be assigned to tail t.
- $FA_t \subseteq F$ is the set of tasks which may be the first task in a line of work assigned to tail t.
- C is the set of conflict events. Each conflict event is characterized by a combination of 10-minute time periods and a collection of adjacent parking spots. A possible conflict is identified at each conflict event by combinations of flights scheduled to depart within the predefined time periods and tails located in adjacent parking spots. It should be noted that there is a possible conflict with each flight for which there are other flight departures within a 10-minute time span beginning with the flight's departure.

3.2. Parameters

- c_t^i is a penalty for operating flight i with tail t. This cost accounts for soft operating restrictions.
- b_1^i is the cost of not covering flight i.
- b_2^i is the cost of not covering soft-maintenance task i. Recall that some maintenance tasks, which do not feature urgent, strictly needed, or important repairs or checks may be postponed.
- $\kappa_t^{i,j}$ is a penalty for each combination of tail t and consecutive tasks i and j.
- λ^c is a penalty for each excess conflict at conflict event c.
- μ_i is the change penalty for not operating flight i using the originally planned fleet type.
- $\widehat{w}_i^p$ is 1 flight i was originally scheduled to be operated by fleet type p, and 0 otherwise.
- $\nu_{i,j}$ is the change penalty for not operating tasks i and j consecutively using the originally planned line of work (where task j follows task i in the line of work).
- $\hat{u}_{i,j}$ is 1 flight i was originally scheduled to precede task j, and 0 otherwise.

3.3. Variables

- $x_i^t \in \{0,1\}$ is 1 if tail t is assigned to task i, and is 0 otherwise.
- $y_{i,j}^t \in \{0,1\}$ is 1 if tail t is assigned to consecutive tasks i and j in a line of work and is 0 otherwise.
- $w_i^p \in \{0,1\}$ is 1 if fleet type p is assigned to task i, and is 0 otherwise.
- $a_i^t \in \{0,1\}$ is 1 if tail t starts a line of work whose first task is i, and is 0 otherwise.
- $s_i \in \{0,1\}$ is 1 if task i is not covered and is 0 otherwise.

- $u_{i,j} \in \{0,1\}$ is 1 if tasks i and j are consecutive in a line of work and is 0 otherwise.
- $o_c \in \mathbb{R}^+$ is the number of conflicts in excess of the maximum number of allowed conflicts at conflict event c.

3.4. Objective function

$$z = \sum_{i\in FF}\sum_{t\in T_i^F} c_t^i x_i^t + \sum_{i\in FF} b_1^i s_i + \sum_{i\in FSM} b_2^i s_i + \sum_{i\in F}\sum_{j\in F_i^+}\sum_{t\in T_i^F\cap T_j^F} \kappa_t^{i,j} y_{i,j}^t + \sum_{c\in C}\lambda^c o_c + \sum_{i\in FF}\sum_{p\in P_i}\mu_i\left|w_i^p - \widehat{w}_i^p\right| + \sum_{i\in F}\sum_{j\in F_i^+}\nu_{i,j}\left|u_{i,j} - \hat{u}_{i,j}\right| \quad (1)$$

The objective function in (1) has a total of seven terms in the following order. Penalties for unsuitable task-tail combinations, costs for not covering flights, costs for not covering soft-maintenance duties, penalties for any task link, and penalties for any conflict over the maximum allowed number. The objective function's last two terms penalize deviations from the originally intended schedule. The first penalizes deviations from the initial fleet type assignment. The second penalizes deviations from the initial line of work.

3.5. Task covering constraints

$$\sum_{p\in P_i} w_i^p + s_i = 1 \qquad \forall i \in F \quad (2)$$

$$\sum_{t\in T_p^P\cap T_i^F} x_i^t = w_i^p \qquad \forall i \in F, p \in P \quad (3)$$

Constraints (2) state that each task is either assigned to one fleet type or unassigned. Constraints (3) say that, if a task is assigned a fleet type, it must be assigned a tail that belongs to that fleet type and is compatible with the task.

3.6. Line-of-work constraints

$$\sum_{i\in FA_t} a_i^t \le 1 \qquad \forall t \in T \quad (4)$$

$$u_{i,j} = \sum_{t\in T_i^F\cap T_j^F} y_{i,j}^t \qquad \forall i \in F, j \in F_i^+ \quad (5)$$

Constraints (4) are constraints on line-of-work initialization. They assign the first task in the line of work to each tail. Constraints (5) define task lines in terms of succession regardless of the allocated tail.

3.7. Task sequencing constraints

$$\sum_{i\in F_j^-\cap F_t^T} y_{i,j}^t + a_j^t = x_j^t \qquad \forall j \in F, t \in T_j^F \quad (6)$$

$$\sum_{j\in F_i^+\cap F_t^T} y_{i,j}^t \le x_i^t \qquad \forall i \in F, t \in T_i^F \quad (7)$$

Task sequencing restrictions are constraints (6) and (7). Constraints (6) are backward sequencing constraints; for each task in a line of work to be allocated to a compatible tail, it must be preceded by another task, unless it is the first task in the line of work. Constraints (7) are forward sequencing constraints; there can be up to one successor for each task in a line of work to be allocated to a compatible tail.

3.8. Other constraints

Owing to space limitations, other constraints are not directly shown here. Seating capacity on each cabin type must be equal to or greater than the number of confirmed reservations. Constraints to ensure that operational restrictions are not broken. Constraints preventing night flights from taking place on two consecutive nights. Constraints stating that no tail can fly consecutive nights in order to ensure that every tail rests overnight at least once every two days. Constraints restricting the number of tails that can be used for each day's schedule to the number that are available. Constraints to ensure that hard maintenance and regular maintenance activities are allocated to the appropriate tail. Constraints stating that daily maintenance duties are conducted at airports where possible. Constraints on the number of conflicts that may occur, limiting the number of conflicts exceeding the airline's overall allowable.

4. Computational experiments

We assessed the model's success using case studies based on real-world examples from Vueling Airlines. The information was given by the airline and represents its operations in Europe in 2019. The data set includes operational schedule details, operating expenses, passenger demand values, the BCN airport apron layout, maintenance capacities, and the available fleet from October 6 to October 10, 2019. The air network features 173 airports spread across Europe, as well as those in Asia and Africa. On a typical day, approximately 700 flights operate throughout the network. Three fleet types were available in this case study: a fleet of A-319s with 141 seats per plane, a fleet of A-320s with 171 seats per plane, and a fleet of A-321s with 200 seats per plane. A series of case studies was suggested to evaluate the model and solution methods for real-world instances. All of the case studies were set in the same time period but had different planning horizons ranging from one to five days. This essentially means that the problem size was different with each case study. The tests were performed on an Intel NUC machine equipped with an Intel Core i7-8559U @ 2.70GHz processor and 2x16GB SO-DIMM DDR4 2400 MHz RAM, running Windows 10 Pro. The models were written in Python 3.7.3 and solved with the commercial solver IBM ILOG CPLEX 12.9.0. Many of the instances tested were either solved to perfection or ran for less than 24 hours.

Table 1 displays the mathematical model size for each of the case studies, with each row representing a different case study. Table 1 also indicates how many flights, maintenance duties, and tails are available in each case study. The number of (discrete) variables, constraints, and nonzero elements were given as model sizes.

We began by solving all the case studies using the branch-and-cut and heuristics approaches given by the commercial solver IBM ILOG CPLEX 12.9.0. Table 2 displays these findings. It contains a case study for each row, which is defined by the number of days in the planning horizon, the number of flights, the number of maintenance duties, and the number of tails. Table 2 also displays the lower bound (L.B.), incumbent solution (I.S.), optimality gap (O.G.), and computational time in seconds for each case study (T.). The lower bound is equal to the solver's highest bound. The incumbent solution is the best solution discovered. The optimality gap is the relative gap between the incumbent solution and the lower bound. The computational time is the amount of time the solver spent running. Except for the case study involving 5 days, all of the case studies were solved to optimality; however, as the problem size grew, the computational time increased exponentially, implying that this solution strategy was unable to produce solutions within a reasonable time if the timeframe to be solved was longer than a few days.

Table 1. Model size for different case studies.

No. of days	Flights	Maintenance tasks	Tails	Variables	Constraints	Non-zero elements
1	682	59	127	117,217	120,342	352,884
2	1,376	77	127	454,550	242,922	1,150,661
3	1,999	86	127	998,446	352,142	2,347,908
4	2,626	95	127	1,729,013	464,960	3,925,464
5	3,279	100	127	2,731,427	581,291	6,048,599

To efficiently solve the problem, we created and implemented an algorithm based on rolling horizon methods (Sethi and Sorger, 1991) to obtain solutions. The Rolling Horizon Algorithm (RHA) is a technique for solving mixed 0-1 deterministic optimization problems that is based on rolling horizon methods. It involves solving a series of integer programming subproblems in which the variables are partitioned into three subsets. The values of the variables in the first subset are fixed to previous solution values, the 0-1 variables in the second subset are held free, and the values of the variables in the third subset are fixed to 0. However, the RHA cannot prove optimality. A particular approach should be taken to demonstrate it. To that end, the solution can be used as an initial solution for another approach to solve the whole problem. While exact methods should be used to ensure optimality, we have empirically discovered that feeding the CPLEX "solution polishing" heuristic with the initial solution obtained by the RHA provides the optimal solution. Table 3 displays the solutions obtained for the case studies with planning horizons of 4 and 5 days. The obtained incumbent solutions are equal to the respective lower bounds in Table 2, indicating that they are optimal. Furthermore, computational times are significantly reduced.

Table 2. Solutions of all the case studies using the branch-and-cut and heuristics approach.

No. of days	Flights	Maintenance tasks	L.B.	I.S.	O.G. (%)	T. (s)
1	682	59	193.050	193.050	0.00	1.05
2	1,376	77	333.025	333.025	0.00	7.88
3	1,999	86	451.525	451.525	0.00	1,200.27
4	2,626	95	599.350	599.350	0.00	47,312.28
5	3,279	100	728.100	18,690.100	2,466.96	86,403.72

Table 3. RHA solution for the case studies featuring planning horizons of 4 and 5 days.

No. of days	Flights	Maintenance tasks	I.S.	O.G. (%)	T. (s)
4	2,626	95	599.350	0.00	279.50
5	3,279	100	728.100	0.00	407.42

Table 4. Comparison of the model solutions with those used by the airline.

	Model			Airline		
No. of days	U.	S.V.	H.V.	U.	S.V.	H.V.
1	0	26	0	0	42	3
2	0	56	0	0	90	7
3	0	80	0	0	126	18
4	0	95	0	0	171	27
5	0	119	0	0	224	35

To determine the quality of the model's solutions, they were compared to the actual solutions implemented by the airline. In this comparison, three major performance indices were examined: the number of unassigned tasks (U.), which were either flights or maintenance tasks, the number of hard constraint violations (H.V.), and the number of soft constraint violations (S.V.). Table 4 shows the comparison, which is made the day before the day of operations, when schedules are ready to be implemented. The case study is described in the first column of Table 5. The other two main columns, Model and Airline, display the key performance indices for the mathematical model's and the airline's solutions, respectively. Note that the solutions provided by the model never violate hard restrictions. Moreover, the number of soft restriction violations is significantly reduced.

Maintenance operations scheduling in airlines is a difficult and complex issue. Maintenance plans are usually prepared in practice based on the expertise of maintenance operators. However, this method is typically time consuming and can result in subpar solutions. Many industries, including the airline industry, are designing better maintenance plans to maximize asset availability and performance (Deng et al., 2020). Predictive maintenance techniques predict when maintenance should be done. It saves money over preventive maintenance since tasks are only done when they are required. The aim of predictive maintenance is to make it simple to schedule corrective maintenance to avoid unexpected failures. Two additional studies were carried out to demonstrate the possible benefits of using a holistic predictive maintenance method. For the two tests, a 5-day planning horizon was selected. In the first experiment, there is insufficient knowledge on maintenance duties for the entire planning horizon, which means that some of them are revealed as time passes. The aim of this environment is to mimic the airline's current operating model, in which a near-perfect predictive maintenance method is currently unavailable. In the second experiment, maintenance duties feature full or perfect details, implying that a perfect predictive maintenance tool is usable. In the first experiment, the mathematical model was solved every day, which means the model must be solved 5 times. The number of disclosed maintenance duties varied for each run of the model, implying that the schedule is not static. It should be noted that the model's 5 runs were also embedded in a rolling horizon approach. In the second experiment, the knowledge was perfect, so the five days could be solved in a single execution. Table 5 shows the results. The first column lists the main performance indicators (KPIs), the second the solution to the first experiment, i.e., the imperfect information scenario, and the third the solution to the second experiment, i.e., the perfect information scenario. The value of the objective function for the incumbent solution (I.S.) in the second row of the table, the number of unassigned tasks (U.) in the third row, and the number of violations of soft constraints (S.V.) in the last row of the table were used to compare the two scenarios. The findings of the perfect information scenario clearly outperform those of the imperfect information scenario.

Table 5. Results for the imperfect and perfect information scenarios.

KPI	Imperfect information scenario	Perfect information scenario
I.S.	1,655.975	1,072.450
U.	6	0
S.V.	125	119

5. Conclusions

We took a novel approach to the tail assignment issue in airlines. The method we devised gathers a broad range of data and offers a basis for generating optimal proposals rather than just feasible solutions. Among the specifics considered are all applicable aircraft maintenance constraints and flight activity requirements. Furthermore, possible conflicts during aircraft taxi operations in aprons are considered, so that departures are optimally planned to prevent multiple aircraft from departing the same place at the same time. We were able to solve real-world instances in short computational times while proving the optimality of the given solutions. The algorithms we created to solve the problem are divided into two stages. Firstly, a feasible solution is found using the rolling horizon process. Secondly, a heuristic-based approach improves the feasible solution obtained. We presented the findings of several computational experiments conducted using data from Vueling Airlines, a major Spanish airline.

Acknowledgements

This research was supported by Project Grants TRA2016-76914-C3-3-P (MINECO), CAS19/00036 (MICIU), and PID2020-112967GB-C33 (AEI), Spain.

References

Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., & Shenoi, R. G. (1998). Flight string models for aircraft fleeting and routing. Transportation science, 32(3), 208-220.

Deng, Q., Santos, B. F., & Curran, R. (2020). A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. European Journal of Operational Research, 281(2), 256-273.

Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1997). Daily aircraft routing and scheduling. Management Science, 43(6), 841-855.

Gabteni, S., & Grönkvist, M. (2009). Combining column generation and constraint programming to solve the tail assignment problem. Annals of Operations Research, 171(1), 61.

Grönkvist, M. (2005). The tail assignment problem (pp. 0328-0328). Chalmers tekniska högskola.

Lan, S., Clarke, J. P., & Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passen

ger disruptions. Transportation science, 40(1), 15-28.

Liang, Z., & Chaovalitwongse, W. A. (2013). A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem. Transportation Science, 47(4), 493-507.

Maher, S. J., Desaulniers, G., & Soumis, F. (2018). The daily tail assignment problem under operational uncertainty using look-ahead maintenance constraints. European Journal of Operational Research, 264(2), 534-547.

Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. Annals of Operations Research, 29(1), 387-415.