

An Optimisation Framework for Airline Fleet Maintenance Scheduling with Tail Assignment Considerations

David Torres Sanchez^a, Burak Boyacı^b, Konstantinos G. Zografos^{b,*}

^a*STOR-i CDT, Lancaster University, Lancaster, LA1 4YX, UK*

^b*Centre for Transport and Logistics (CENTRAL), Lancaster University Management School, Lancaster, LA1 4YX, UK*

Abstract

Fierce competition between airlines has led to the need of minimising the operating costs while also ensuring quality of service. Given the large proportion of operating costs dedicated to aircraft maintenance, cooperation between airlines and their respective maintenance provider is paramount. In this research, we propose a framework to develop commercially viable and maintenance feasible flight and maintenance schedules. Such framework involves two multi-objective mixed integer linear programming (MMILP) formulations and an iterative algorithm. The first formulation, the airline fleet maintenance scheduling (AMS) with violations, minimises the number of maintenance regulation violations and the number of not airworthy aircraft; subject to limited workshop resources and current maintenance regulations on individual aircraft flying hours. The second formulation, the AMS with tail assignment (TA) allows aircraft to be assigned to different flights. In this case, subject to similar constraints as the first formulation, six lexicographically ordered objective functions are minimised. Namely, the number of violations, maximum resource level, number of tail reassignments, number of maintenance interventions, overall resource usage, and the amount of maintenance required by each aircraft at the end of the planning horizon. The iterative algorithm ensures fast computational times while providing good quality solutions. Additionally, by tracking aircraft and using precise flying hours between maintenance opportunities, we ensure that the aircraft are airworthy at all times. Computational tests on real flight schedules over a 30-day planning horizon show that even with multiple airlines and workshops (16000 flights, 529 aircraft, 8 maintenance workshops) our solution approach can construct near-optimal maintenance schedules within minutes.

Keywords: Airline Maintenance Scheduling, Tail Assignment, Multi-objective Mixed Integer Linear Programming

1. Introduction

There are a number of operational decisions associated with airlines, from ticket prices to flight times, crew rosters, and aircraft maintenance. When making these decisions, airlines have to take into account their own economic interests influenced by demand, costs, and sometimes even the actions of their competitors. In such a competitive environment, airlines aim to minimise their operating costs while providing competitive services. Significant proportion of operating costs are dedicated to maintenance. For instance, 20.5% of the average direct operating cost per medium-haul trip are dedicated to maintenance on an Airbus A330-200 ([Aircraft Analysis & Fleet Planning, 2005](#)). Therefore, it is of paramount importance to develop decision making tools that will allow airlines to optimise their aircraft maintenance decisions.

Maintenance types are classified according to: short, medium and long-term interventions. Short-term or line maintenance does not require modelling or advanced planning as they are carried out as standard procedures at airport gates. Medium and long-term maintenance interventions include:

*Corresponding author

Email addresses: d.torressanchez@lancaster.ac.uk (David Torres Sanchez), b.boyaci@lancaster.ac.uk (Burak Boyacı), k.zografos@lancaster.ac.uk (Konstantinos G. Zografos)

1. Airframe checks (A, B, C and D);
2. Engine performance restoration (EPR) and life limited parts replacement (ELR);
3. Landing gear overhaul (LG), and;
4. Auxiliary power unit (APU) performance restoration.

Civil Aviation Authorities Regulations impose that maintenance has to be performed after a certain number of months (MO), flying hours (FH), or flight cycles (FC), at certified maintenance workshops. In the medium-term, A checks have to be performed every 80-100 FH (every 7 to 9 days), requiring 10-20 man-hours, while B checks typically occur every 500-600 FH (every two months), requiring 100-300 man-hours (Department for BIS, 2016). However, in practice, Type B checks are included as part of a longer A check, or a bundle of A checks (Qantas, 2016). Long-term maintenance, including C and D checks, LG, EPR, ELR, and APU are performed once every 1–6 years and can last over 10 days (Ackert, 2011). It is worth noting that there is a large variability in the duration of maintenance checks due to the fact that different aircraft types have different maintenance requirements.

Table 1 shows the frequency of the four airframe checks for various aircraft types. As one would expect, the time between checks increases for more modern aircraft and B checks disappear, being contained in longer A checks. Maintenance is performed before any of the three criteria (MO, FH or FC) is met. For instance, for the B737-200, a C check is performed after 18 MO, 6000 FH, or 3000 FC, whichever occurs first. The justification for this practice is to ensure coverage of overused aircraft operating short-haul flights. In these cases, FC are accumulated faster than FH (Cook and Tanner, 2008). Moreover, usual frequencies of long-term maintenance, for the A320, for instance, are 13500 FC for both the EPR and the ELR, 120 MO/20000 FC for the LG, and 75000 FH for the APU (Ackert, 2011).

Table 1: Typical maintenance frequencies in calendar months (MO), flying hours (FH), or flight cycles (FC) (Cook and Tanner, 2008; Martins, 2016).

Aircraft	A check	B check	C check	D check
B737-300	275 FH	825 FH	18 MO	48 MO
B737-400	275 FH	825 FH	18 MO	48 MO
B737-500	275 FH	825 FH	18 MO	48 MO
B737-800	500 FH	n/a	4000-6000 FH	96-144 MO
B757-200	500-600 FH	n/a	18 MO/6000 FH/3000 FC	72 MO
F100	500 FH	n/a	5000 FH	12000 FH
B767-300ER	600 FH	n/a	18 MO/6000 FH	72 MO
B747-400	600 FH	n/a	18 MO/7500 FH	72 MO
A319	600 FH	n/a	18-20 MO/6000 FH/3000 FC	72 MO
A320	600 FH	n/a	18-20 MO/6000 FH/3000 FC	72 MO

For maintenance to be performed effectively, there is an essential underlying process, airline fleet maintenance scheduling (AMS). The AMS problem deals with the construction of a schedule that minimises maintenance costs, resource usage, and the disruption to airline operations, while satisfying current safety regulations by different civil aviation authorities (Sriram and Haghani, 2003). Given the frequency of the maintenance checks, the decision horizon for medium-term maintenance should be a month, and, at least, six months for long-term maintenance.

A challenge for the AMS problem is to allocate maintenance-related resources in a cost-effective fashion. These resources are geographically dispersed throughout distant and distinct maintenance workshops. Some examples of resources include limited specialised tools, spare parts, and certified technicians. Additionally, regulated checks depend on the state of the aircraft and employ different resources.

The major contribution of this paper is the framework that deals with the requirements introduced by 30-day planning horizon instances, with multiple airlines and workshops, and tight resource availabilities. Such framework can be broken down into three steps. Firstly, the preprocessing step identifies maintenance opportunities (MOPs). These opportunities arise when flights have large turnaround times during which aircraft can be maintained. This preprocessing allows us to formulate and solve the problem efficiently. Next, in the case when not enough MOPs are found and maintenance regulations are violated, we identify when these occur and we reassign aircraft to different flights to generate more MOPs. In other words, we re-solve the tail assignment (TA) problem. Lastly, to preserve

tractability and improve the quality of solutions, we provide a two-stage iterative algorithm. The first stage selects a conflicting period which identifies the smallest number of flights whose reassignment might lead to a maintenance feasible solution. Such period starts from the maintenance regulation violation, and ends at the next available MOP. The second stage makes the timeline more granular to improve resource allocation. Further contributions introduced by this work are: the presentation of a new type of multi-objective optimisation formulation for the AMS which copes with single and multi-workshop cases, the focus on workshop resource allocation, the consideration of different fleet types and their respective maintenance requirements, and, the introduction of a solution framework that solves problems of realistic size in short time. Showing the potential of this framework which promotes cooperation between airlines.

The remainder of this paper is organised as follows: Section 2, discusses the relevant literature. Section 3 presents the proposed modelling approaches and the corresponding underlying concepts. The first model, minimises the number of maintenance regulation violations, adhering to the criteria presented in Table 1. The second model solves a reduced TA problem within the AMS considering violations. Section 4 presents the solution methodology and the model application. Computational experiments are given in Section 5, while Section 6 summarises the conclusions and provides recommendations for future research.

2. Literature Review

The airline planning process or airline scheduling problem involves several stages: flight scheduling, sometimes referred to as schedule design; fleet assignment, which assigns fleet types to flights (Hane et al., 1993); TA, sometimes called aircraft routing or aircraft rotation and involves assigning individual aircraft to flights (Clarke et al., 1997); maintenance scheduling (MS); and, finally, crew scheduling. These were traditionally modelled by formulating each stage separately and solving them sequentially, i.e. the output of one stage is the input for the next. Figure 1 presents the order of solving the different types of problems identified above (solid lines), along with some common feedback loops (dashed lines) and the associated typical planning horizons.

The sequential modelling and solution of the airline scheduling problem does not take into consideration the restrictions of the subsequent problems. The benefit of the sequential approach is the reduction in computational complexity. Even though the sequential feedback system is a close approximation, the solution can be improved by modelling the interdependence of each stage in an integrated model (Cordeau et al., 2001). Integrated models have been developed to provide better quality results and consider the combination of two or more stages into a single problem (Desaulniers et al., 1997; Clarke et al., 1997; Barnhart et al., 1998; Cohn and Barnhart, 2003; Sriram and Haghani, 2003; Mercier et al., 2005; Sarac et al., 2006; Liang and Chaovalitwongse, 2013; Safaei and Jardine, 2018). In particular, MS is frequently contained within the TA, in which case it is called aircraft maintenance routing (AMR) problem (Gopalan and Talluri, 1998).

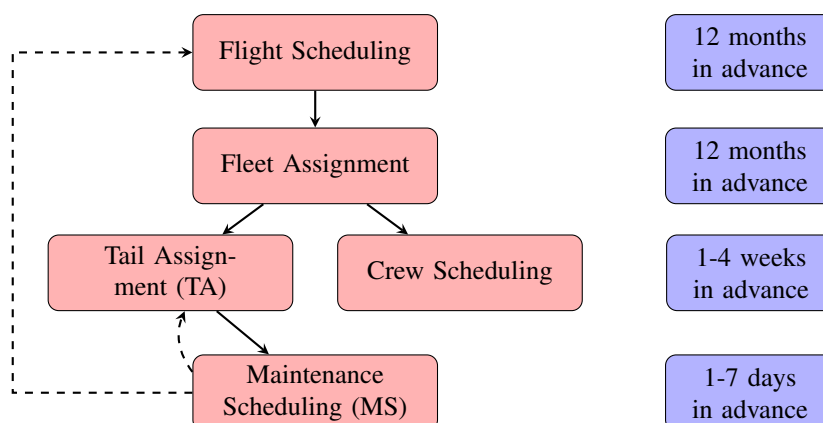


Figure 1: Stages of the airline scheduling problem.

The most common types of mixed integer programming formulations for the integrated airline scheduling problem can be classified into three groups,

String-based models: a type of formulation that models the problem using strings, i.e. sequences of connected flights that begin and end at a maintenance workshop, and, that satisfy flow balance and maintenance regulations (Desaulniers et al., 1997).

Time-Space Network (TSN) models: in a TSN network, each airport is represented by a time line showing the planning horizon. Nodes show every departure/arrival at the corresponding airport time line and arcs show flights and connections (Hane et al., 1993). For example, in Figure 2, there are two airports, *A* and *B*. Solid arcs represent scheduled flights, while grey dashed arcs represent deadhead flights. A flight path between the two timelines, shown in blue, starts at airport *A* at time period 1 (node A_1), flies to airport *B* arriving at time period 2 (node B_2), then is grounded at airport *B* until time period 3 (node B_3), and so on;

Multi-Commodity Network Flow (MCNF) models: based on a fleet-flow time-space network (layered TSN models), each aircraft represents separate commodities and flow has to be preserved. Formulations of this type typically include constraints regarding capacities (passengers and fleet) and flow conservation (aircraft, flight, and airport) (Levin, 1971).

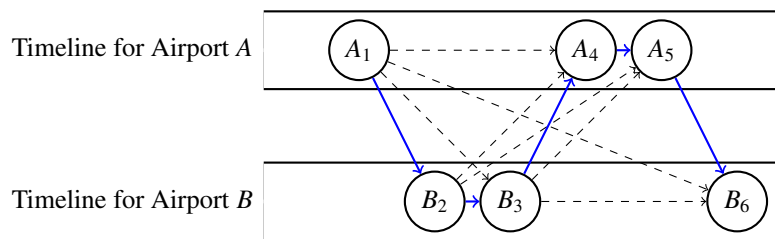


Figure 2: Time-Space Network (TSN) example.

The first string-based formulations were introduced by Desaulniers et al. (1997), for the FA problem, and Barnhart et al. (1998), for the TA problem. They both implemented a branch and bound scheme as their solution method. Even though they included maintenance regulations, workshop resources or FH were not considered. Sarac et al. (2006); Cohn and Barnhart (2003); Papadakos (2009) used the same type of formulation for the AMR problem. Papadakos (2009) produced a computational case study for a medium-sized data set (700 flights, 167 aircraft). The heuristic algorithm used required 16 hours to solve the problem under consideration. Sarac et al. (2006) implemented a branch-and-price algorithm and used legal remaining FH to influence decisions. Aside from the excessive solution times, string-based formulations are not capable of generating all strings even for small instances. Furthermore, these models do not include resource usage for maintenance activities.

The TSN formulation, introduced by Hane et al. (1993), has been used widely within the integrated airline scheduling framework, (Clarke et al., 1997; Hicks et al., 2005; Orhan et al., 2012; Haouari et al., 2013; Liang and Chaovaitwongse, 2013). However, given that TSNs do not allow individual aircraft to be tracked, aggregated maintenance constraints are implemented. This means that models are forced to, for instance, minimise total weekly maintenance operations. More recently, Safaei and Jardine (2018) examined the AMR problem with generalised maintenance constraints and legal remaining FH considerations. They used test instances for a single airline. The computational study, for a relatively large data set (7 days, 772 flights, 18 aircraft), solutions show financial impact but computational times are not provided.

Using the MCNF formulation, introduced by Levin (1971), the problem can be solved using column generation (Yan and Tseng, 2002; Sriram and Haghani, 2003; Mercier et al., 2005). Particularly, Sriram and Haghani (2003), presented some influential work, based on the TA formulation by Feo and Bard (1989), which solved the AMR incorporating A and B checks. They used a heuristic algorithm that solved a small test instance (58 flights, 75 airports) in 5 minutes.

All the publications mentioned thus far are limited to daily or weekly schedules and assume cyclical repetitions of the flights and hence maintenance operations. [Khaled et al. \(2018\)](#), however, considered individual maintenance requirements for a 30-day plan. They used an improved TSN formulation for the AMR problem, which allowed them to effectively schedule A checks, constrained by individual aircraft legal remaining FH. They assumed that maintenance is generally performed at night. Due to the type of formulation, solution times increased noticeably for a large number of flights (timed out at 3 hours for 1494 flights and a single airline). [Li et al. \(2016\)](#), dealt with the AMR problem for fighter jets. This formulation, which relied on a single workshop assumption, also employed legal remaining FH to determine the frequency of maintenance. Further assumptions included: no resource considerations, only one type of maintenance, and the disregard of aircraft “health” at the end of the planning horizon. The test set considered is of 25 days and 200 aircraft but no solution times are provided. In addition, the timeline is discretised by splitting each day into two 12 hour intervals, clearly, this incurs a huge loss in accuracy.

In different context, we find formulations that account for long-term planning horizons; specifically, in the resource constrained project scheduling problem (RCPSP) literature. The RCPSP is a generalisation of machine scheduling problem where jobs are scheduled according to some predefined order, or precedence, subject to different resource demands and capacity constraints. Usually, the objective is to minimise the duration of the project (collection of ordered jobs), commonly, under a non-preemption assumption (jobs have to be processed fully) ([Koné et al., 2011](#); [Brucker and Knust, 2012](#); [Kopanos et al., 2014](#); [Naber, 2017](#)). The reader may refer to a recent and thorough review article by [Habibi et al. \(2018\)](#). On the long-term RCPSP, [Koné et al. \(2011\)](#) proposed two formulations which they named “event-based RCPSP” formulations. Events correspond to start and/or end times of activities. Since their formulations involve fewer variables than the formulations indexed by time, they have the capacity to deal with longer planning horizons.

Table 2: Summary of selected literature. MS = inclusion of maintenance scheduling, FH = use of legal remaining flying hours, RC = workshop resource considerations, H = time horizon (days), Sched. Type = scheduling type, Airlines = number of airlines considered, W = number of workshops considered, F = number of flights (in largest test instance), A = number of aircraft (in largest test instance), T = computational time (of largest test instance).

Article	MS	FH	RC	Sched. Type	Airlines	W	H	F	A	T
Desaulniers et al. (1997)	×	×	×	cyclical	single	1	1	383	91	1 h
Barnhart et al. (1998)	✓	×	×	cyclical	single	1	7	1124	89	10 h
Sriram and Haghani (2003)	✓	×	×	cyclical	single	20	7	rand.	13	4.5 h
Mercier et al. (2005)	✓	×	×	cyclical	two	NS	1	707	143	13 h
Sarac et al. (2006)	✓	✓	✓	cyclical	single	5	1	175	32	2 h
Papadacos (2009)	✓	×	×	cyclical	single	6	7	705	167	16 h
Haouari et al. (2013)	✓	×	×	cyclical	single	16	1	344	138	10 s
Liang and Chaovalitwongse (2013)	✓	×	×	cyclical	single	6	7	1780	110	4 h
Khaled et al. (2018)	✓	✓	×	adaptive	single	9	30	1494	40	3 h
Present paper	✓	✓	✓	adaptive	single	1	30	3869	49	1.2 h
Present paper	✓	✓	✓	adaptive	multiple	8	30	16000	529	1.8 h

As can be seen in the summary of the literature in Table 2, no publications have addressed the long-term AMS problem with multiple airlines, workshops and resource considerations, while providing fast solution times. Additionally, we take into account individual aircraft maintenance requirements and their respective flight operations. The key contributions of the present paper are,

Reassignment. We include the option of reassigning some flights to obtain longer maintenance opportunities.

Efficient resource allocation. Maintenance models do not always consider the different resources available throughout maintenance workshops. Therefore, we incorporate workshop resource restrictions.

Individual aircraft considerations. Different aircraft may have different maintenance duration and requirements, and different accumulation of FH. Hence, we include this in our model.

Long-term planning horizon. Short-term, or operational planning, is not suitable for most aircraft maintenance. Further, personnel and equipment hire are significantly expensive. Thus, it is extremely useful for maintenance operators to plan longer in advance.

Two-stage iterative algorithm. We employ a two-stage algorithm that provides good solutions for large instances in reasonable computational time.

Single and multi-workshop tests cases. Our generic framework, allows us to model both single and multi-workshop cases.

3. The Proposed Modelling Approach

In order to be able to adequately model the problem, we first state the necessary assumptions, highlight the concepts used and define the appropriate notation. After this, we propose two multi-objective mixed integer linear programming (MMILP) formulations that schedule different types of maintenance for a medium/long-term planning horizon e.g. airframe checks A and C. To check if regulation requirements are being fulfilled, we employ different maintenance requirement (MR) variables for each type of maintenance and aircraft type, which vary with FH.

In the first formulation, given an input flight schedule, we aim to determine whether a feasible maintenance schedule exists. We call this the AMS formulation. If a feasible maintenance schedule does not exist, we seek to minimise the number of infeasibilities, or violations. Such violations represent the cases when the limit imposed by the regulations on the FH are exceeded. To minimise the violations, we introduce the second formulation which extends the AMS formulation to account for an appropriate TA problem.

3.1. Assumptions

To ease the formulation of the problem, we make some modelling assumptions.

1. Maintenance can only be performed in the pre-identified MOPs;
2. Maintenance can be done at any workshop, if the aircraft has a MOP there; unless otherwise specified by an airline;
3. Maintenance cannot be preempted;
4. To extend MOPs, we may reassign flights to different aircraft, i.e. we may modify the airlines' preferred TA;
5. Resources can be shared amongst different airlines at the maintenance workshops.

3.2. Concepts

The integrated airline scheduling literature reveals that the models used for short-term planning are not easily scalable for our 30-day planning horizon. Additionally, not much attention is paid to either resource usage for maintenance activities or aircraft health-state monitoring. However, as mentioned in the literature review section, [Koné et al. \(2011\)](#) presented a formulation for the long-term resource constrained project scheduling problem (RCPS) that challenged the classical discretisation of time. Instead, they index the variables using some pre-defined "events".

[Koné et al. \(2011\)](#), independently formalised the same idea as [Sousa and Wolsey \(1992\)](#), and refer to this type of formulation as an "event-based RCPS". Events represent either the start or the end of an activity. Compared to traditional time indexation, their formulations involve considerably fewer variables, therefore, being ideal for problems with long planning horizons. [Koné et al. \(2013\)](#) extended the formulation to account for non-renewable resources. More authors have used a similar indexation of continuous time. [Naber \(2017\)](#) focused on removing the assumption regarding fixed resources per activity in the RCPS by allowing flexibility of resource usage.

In the present paper, events represent turnaround times, the time between arrival and departure, at a certain maintenance workshop where the total time is sufficient to perform at least one type of maintenance. We regard these as MOPs. Thus, provided with a flight schedule, MOPs are easily identified and pose no restrictive assumption on the aircraft or airlines considered. For more information on how these are created, see Section 4.1.

3.3. Airline Fleet Maintenance Scheduling with Violations

The AMS with violations model checks, subject to regulations, whether a feasible maintenance schedule exists given a flight schedule. Violations of regulations, contrary to most other works, are not included as constraints but as an objective which we try to minimise. Hence, the impasse that infeasibilities bring is avoided and more conclusions can be inferred about the process. First, the necessary sets, parameters and variables are defined, then we introduce the MMILP formulation.

3.3.1. Notation

To formulate the problem, we introduce the following notation. The set of all aircraft is denoted by \mathcal{K} and indexed with k . In order to distinguish aircraft by type and airline, we introduce the set \mathcal{T} with function $t: \mathcal{K} \rightarrow \mathcal{T}$; $t(k)$ maps a specific aircraft $k \in \mathcal{K}$ to its corresponding fleet type and airline. Each aircraft has its own corresponding MOPs contained in the set MOP_k which we can index with j . Moreover, we can subdivide each MOP into time intervals. These provide an alternative discretisation of time and represent either the start or end time of some MOP. All time intervals are contained in the set \mathcal{I} , indexed by i . Each element i , has a start and end time, denoted with st^i and et^i respectively, and a maintenance workshop W^i . Some common maintenance checks are collected in the set \mathcal{C} , indexed by c . Resources are contained in the set R . The demand for resource $r \in R$ varies per check $c \in \mathcal{C}$, this is denoted by b_{rc} . Moreover, the duration of maintenance may vary within aircraft types. Hence, for an aircraft k , of type $t(k)$, we define the duration of a single check $c \in \mathcal{C}$ to take $\Delta_{t(k),c}$ time units to complete.

For convenience, we can identify the following subsets. Let \mathcal{I}_k be the subset of intervals where aircraft k is available for maintenance. Similarly, let \mathcal{K}^i be the subset of aircraft available for maintenance at interval i . The resources available at the maintenance workshop of interval i , W^i , are contained in the set $R_{W^i} \subseteq R$.

Each MOP can be represented by a set of consecutive intervals. For $j \in MOP_k$, we can identify the corresponding intervals as $i \in MOP_k^j \subseteq \mathcal{I}_k$. This idea is illustrated in Figure 3, where an example for an aircraft k with three MOPs is shown. The numbering indicates the interval number. Each of MOP is labelled with a different $j = 1, 2, 3$. It is worth noting that every interval is assigned to exactly one MOP. We can identify the interval sets $\mathcal{I}_k = \{1, 2, 3, 7, 8, 11, 12\} \subset \mathcal{I} = \{1, \dots, 12\}$. Thus, given the definition of MOPs, intervals in $\mathcal{I} \setminus \mathcal{I}_k = \{4, 5, 6, 9, 10\}$ must belong to another aircraft (not pictured). Moreover, if i is part of a MOP for aircraft k , $i \in \mathcal{I}_k$, we can say that aircraft k is not flying at interval i , hence, is available for maintenance. Conversely, if i is not part of any MOP for aircraft k , $i \notin \mathcal{I}_k$, we can say that aircraft k is flying at interval i .

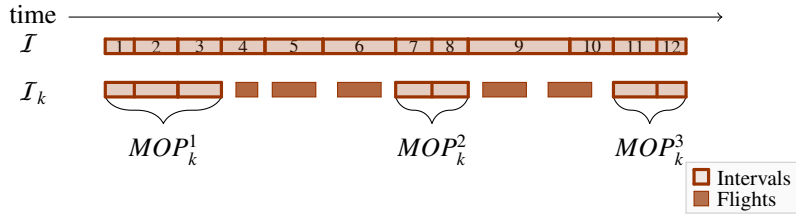


Figure 3: Timeline showing the deconstruction of three MOPs into sets of consecutive intervals.

In order to employ FH accurately in the model, we introduce two additional parameters. Let FH_k^i be the FH for aircraft k at interval i , and, $L_{t(k),c}$ be the limit on FH imposed by regulations for aircraft of type $t(k)$ and check c . In order to update the FH parameter, we consider the flights between consecutive MOPs. For a given interval i and aircraft k , interval i is bounded by

$$\sup\{MOP_k^{j-1}\} < i \leq \sup\{MOP_k^j\}, \quad (1)$$

for some MOP_k^j provided that $j \neq 1$, $i > \sup\{MOP_k^1\}$ and $i \leq \sup\{MOP_k^j\}$ ($J = |MOP_k|$) (proof provided in [Appendix A](#)). That is, provided that the interval under consideration starts after the end of the first MOP and before the end of the last MOP. If this is the case, we define FH_k^i to consider only the flights that aircraft k operates between MOP_k^{j-1} and MOP_k^j . Otherwise, if i starts before the end of the first MOP, i.e. $i \leq \sup\{MOP_k^1\}$, we define FH_k^i to consider only the flights that aircraft k operates between the beginning of the planning horizon and MOP_k^1 . Conversely, if i starts after the end of the last MOP, i.e. $i > \sup\{MOP_k^J\}$, we define FH_k^i to consider only the flights that aircraft k operates

between MOP_k^j and the end of the planning horizon. In Figure 3, if $i \in \{7, 8\} = MOP_k^2 \subset \mathcal{I}_k$, or $i \in \{4, 5, 6\} \notin \mathcal{I}_k$, we have $\sup\{MOP_k^1\} < i \leq \sup\{MOP_k^2\}$, then, FH_k^i considers the flights between MOP_k^1 and MOP_k^2 . If $i \in \{9, 10\} \notin \mathcal{I}_k$, we have $\sup\{MOP_k^2\} < i \leq \sup\{MOP_k^3\}$, then, FH_k^i considers the flights between MOP_k^2 and MOP_k^3 .

Using the previous two parameters, we can define the deterioration that a certain number of FH incurs on an aircraft for a specific check type. For an interval i , check c and aircraft k , the deterioration parameter is defined as,

$$DR_{kc}^i = \frac{FH_k^i}{L_{t(k),c}}. \quad (2)$$

In the next subsection, we provide the notation needed to describe the proposed mathematical model. However, it is useful to discuss two of the variables in detail. To account for the legal remaining FH, we define a variable, $w_{kc}^i \in [0, 1]$, that tracks the MR at the beginning of interval i , for check type c and aircraft k . Where a value close to 0 represents that no maintenance is required, and a value close to 1 represents that maintenance is urgently required. This variable is updated by using the deterioration parameter DR_{kc}^i . We define a binary variable to identify regulation violations, v_{kc}^i gets the value 1 if the regulation for check c is violated at interval i by aircraft k , while it is 0 otherwise.

3.3.2. Definitions

Sets

\mathcal{C} : Set of checks indexed by c ;

\mathcal{I} : Set of intervals indexed by i ;

\mathcal{I}_k : Set of intervals where aircraft k is available for maintenance, $\mathcal{I}_k \subseteq \mathcal{I}$;

\mathcal{K} : Set of all aircraft indexed by k ;

\mathcal{K}^i : Set of aircraft available for maintenance at interval i , $\mathcal{K}^i \subseteq \mathcal{K}$;

MOP_k : Set of MOPs for aircraft k , indexed by j ;

MOP_k^j : Set of intervals that constitute the j -th MOP for aircraft k , $MOP_k^j \subseteq \mathcal{I}_k$;

\mathcal{R} : Set of resources indexed by r ;

R_{W^i} : Set of resources available at the workshop of interval i , W^i , $R_{W^i} \subseteq \mathcal{R}$;

\mathcal{T} : Set of aircraft types with $t: \mathcal{K} \rightarrow \mathcal{T}$.

Parameters

b_{rc} : Demand of resource r to process check c ;

DR_{kc}^i : Deterioration (per flying hour) at interval i for check c and aircraft k ;

$\Delta_{t(k),c}$: Duration of check c for an aircraft of type $t(k)$;

st^i / et^i : Start/end time of interval $i \in \mathcal{I}$.

Variables

B_r^i : The capacity for resource r at interval i .

w_{kc}^i : A continuous variable with values between 0 and 1 to represent the MR for aircraft k for check c at the beginning of interval i . 0 means that the aircraft requires no maintenance, 1 means that the aircraft requires maintenance urgently.

m_{kc}^i : 1, if a maintenance check c for aircraft k starts/continues at the beginning of interval i ; 0, otherwise.

v_{kc}^i : 1, if regulation for check c is being violated at interval i for aircraft k ; 0, otherwise.

z_{kc}^i : 1, if there is a change between the consecutive variables, m_{kc}^i and m_{kc}^{i-1} ; 0, otherwise.

3.3.3. Formulation

Model 3.1. Interval MMILP formulation for AMS with violations.

$$\min \sum_k \sum_c \sum_i v_{kc}^i \quad (3)$$

$$\min \sum_k \sum_c w_{kc}^{\sup\{\mathcal{I}_k\}} \quad (4)$$

Subject to

Maintenance Requirement

$$v_{kc}^i = 0 \quad \forall k, c, i \notin \mathcal{I}_k; \quad (5)$$

$$m_{kc}^i = 0 \quad \forall k, c, i \notin \mathcal{I}_k; \quad (6)$$

$$w_{kc}^i \geq w_{kc}^{i-1} + DR_{kc}^i (1 - m_{kc}^i) - m_{kc}^i - v_{kc}^i \quad \forall k, c, i \neq \inf\{\mathcal{I}\}; \quad (7)$$

$$w_{kc}^{\sup\{MOP_k^j\}} \leq DR_{kc}^{\inf\{MOP_k^{j+1}\}} \quad \forall k, c, j \neq |MOP_k|; \quad (8)$$

Maintenance

$$\sum_c m_{kc}^i \leq 1 \quad \forall k, i \in \mathcal{I}_k; \quad (9)$$

$$\sum_{i \in MOP_k^j} (et^{i'} - st^{i'}) m_{kc}^{i'} \geq \Delta_{t(k),c} m_{kc}^i \quad \forall k, c, j, i \in MOP_k^j; \quad (10)$$

Transitivity Constraints

$$z_{kc}^{\inf\{MOP_k^j\}} \geq m_{kc}^{\inf\{MOP_k^j\}} \quad \forall k, c, j; \quad (11)$$

$$z_{kc}^i \geq m_{kc}^i - m_{kc}^{i-1} \quad \forall k, c, j; i, i-1 \in MOP_k^j; \quad (12)$$

$$\sum_{i \in MOP_k^j} z_{kc}^i \leq 1 \quad \forall k, c, j; \quad (13)$$

Resources

$$\sum_{k \in \mathcal{K}^i} \sum_c b_{rc} m_{kc}^i \leq B_r^i \quad \forall i, r \in R_{Wi}; \quad (14)$$

Variables

$$w_{kc}^i \in [0, 1] \quad \forall k, c, i; \quad (15)$$

$$B_r^i \in \mathbb{R}^+ \quad \forall k, i, r \in R_{Wi}; \quad (16)$$

$$m_{kc}^i, z_{kc}^i, v_{kc}^i \in \{0, 1\} \quad \forall k, c, i. \quad (17)$$

The proposed formulation is a MMILP with two lexicographically ordered objective functions. The functions involved minimise the following objectives (in order of importance), the number of violations, and the total MR at the last interval. Objective 3 minimises the number of regulation violations. Objective 4 minimises the total MR at the end of the planning horizon. Recall that a value close to 0 for the w_{kc}^i variable indicates that the aircraft requires no maintenance; hence, minimising $w_{kc}^{\sup\{\mathcal{I}_k\}}$ corresponds to minimising the amount of maintenance required by each individual aircraft at the end of the planning horizon.

The first two MR constraints 5 and 6 ensure that neither a violation nor maintenance occur when an aircraft is operating flights. Precisely, constraints 5 and 6 ensure that if aircraft k is flying at interval i , i.e. $i \notin \mathcal{I}_k$, then, for any check c , neither a violation nor maintenance intervention may occur.

Constraints 7 enforce a recurrence relation for the MR variable. For time interval i , aircraft k , and check type c , we update the MR variable depending on whether or not aircraft k is flying at interval i . If aircraft k is not flying at interval i , i.e. $i \in \mathcal{I}_k$ and $i \in MOP_k^j$ for some j ; the current MR, w_{kc}^i , is updated using the previous MR, w_{kc}^{i-1} , plus the appropriate deterioration, DR_{kc}^i (the deterioration incurred by the flights between MOP_k^{j-1} and MOP_k^j , as

defined in equation 2), or drops to 0 if either a violation or maintenance occur. Please recall that the first priority objective minimises the number of violations. On the other hand, if aircraft k is flying at interval i , i.e. $i \notin \mathcal{I}_k$, then, by constraints 5 and 6, $m_{kc}^i = 0$ and $v_{kc}^i = 0$, hence, the recurrence relation is enforced.

Constraints 8 ensure that the MR variable at the end of a MOP stays within the regulation limits (captured within the deterioration parameter) at least until the next MOP. That is, for a given aircraft k , check type c and MOP j (with $j \neq |MOP_k|$), the MR at the end of the MOP, $w_{kc}^{\sup\{MOP_k^j\}}$, should remain feasible to operate the upcoming flights between MOP_k^j and MOP_k^{j+1} . Using the definition of the deterioration parameter, this is encapsulated in $DR_{kc}^{\inf\{MOP_k^{j+1}\}}$, since, by equation 1, $\sup\{MOP_k^j\} < \inf\{MOP_k^{j+1}\} \leq \sup\{MOP_k^{j+1}\}$.

Constraints 9 and 10 enforce the maintenance restrictions. Constraints 9 assure that no more than one maintenance type is scheduled for the same interval. Constraints 10 guarantee that the aircraft is available for the minimum time required for each maintenance type. More precisely, the sum of the duration of consecutive intervals has to be greater than the minimum prespecified duration of the check.

Transitivity constraints, 11, 12, and 13, ensure that if we decide to maintain in MOP_k^j , preemptions are not allowed (proof provided in Appendix A). Constraints 11 initialise the auxiliary variable using the first interval in the MOP. Constraints 12 establish that when a maintenance starts, i.e. the difference between consecutive maintenance variables is 1, the auxiliary variable is 1. Constraints 13 ensure that at most one auxiliary variable is 1, or equivalently we cannot start a maintenance more than once. Hence, wherever we terminate maintenance, all auxiliary variables thereafter must be 0.

Constraints 14 ensure that a maintenance intervention of some type is only scheduled if there are sufficient resources available at the workshop. The total number of checks over all aircraft present at a given interval cannot exceed the capacity for each resource. The last four constraints 15 – 17 define the domains of the variables.

3.4. Airline Fleet Maintenance Scheduling with Tail Assignment

In this section we extend the AMS formulation previously discussed, to include reassignment variables for the periods where regulations are being violated. In order to determine where this occurs, we solve Model 3.1, and identify which violation variables, v_{kc}^i , have the value 1. Using this information we efficiently select reassignable and preassigned flights which allow us to solve the joint AMS and TA problem.

3.4.1. Notation

To formulate the AMS with TA, we first expand the notation introduced for Model 3.1. All flights are contained in the set \mathcal{F} , which we can index with f . More precisely, it contains flight legs (sequence of multiple flights) between MOPs. Also, let \mathcal{I}^f be the subset of intervals at which flight $f \in \mathcal{F}$ occurs, and \mathcal{K}^f be the subset of aircraft free to operate flight f .

Additionally, for a given interval i , we can identify the set of flights that either, depart at the end of interval i , or, arrive at the start of interval i . Let us denote the departure set with

$$\mathcal{F}_{dep}^i = \{f : f \in \mathcal{F}, \inf\{\mathcal{I}_f\} = i\},$$

and, the arrival set with

$$\mathcal{F}_{arr}^i = \{f : f \in \mathcal{F}, \sup\{\mathcal{I}_f\} = i\}.$$

In order to control the number of reassignment variables, we can identify the subsets of flights which are *reassignable* and those which are fixed or *preassigned*. Such that, if a flight is reassignable, then we can reassign it to another aircraft; in contrast, if a flight is preassigned, then it is operated by the preassigned aircraft. Let $\mathcal{F}_R \subset \mathcal{F}$ and $\overline{\mathcal{F}}_R \subset \mathcal{F}$ be two (disjoint) subsets (with $\mathcal{F}_R \cup \overline{\mathcal{F}}_R = \mathcal{F}$), that contain those flights which are reassignable and preassigned, respectively. Also, let O^f for flight $f \in \overline{\mathcal{F}}_R$, with $O^f \subseteq \mathcal{K}^f$, contain the aircraft preassigned to flight f . To preserve efficiency and minimise the changes in the airlines' preferred TA, instead of setting all flights to be reassignable, we select an appropriate subset of flights. More details on the selection process can be found in Section 4.2.1.

We can redefine the deterioration parameter in terms of flights, DR_{kc}^f , which represents the deterioration for the operation of flight f for check c and aircraft k . Similarly, as for the previous model,

$$DR_{kc}^f = \frac{FH^f}{L_{t(k),c}},$$

where FH^f are the FH that correspond to flight f . Lastly, the violation variable can also be expressed in terms of flights, v_{kc}^f gets the value 1 if a regulation is being violated before flight f for aircraft k and check c , while it is 0 otherwise.

3.4.2. Definitions

Sets

\mathcal{F} : Set of all flights indexed by f ;

\mathcal{F}_R : Set of reassignable flights, $\mathcal{F}_R \subset \mathcal{F}$;

$\overline{\mathcal{F}}_R$: Set of preassigned flights, $\overline{\mathcal{F}}_R \subset \mathcal{F}$;

\mathcal{F}_{dep}^i : Set of flights which are scheduled to depart at the end of interval i ;

\mathcal{F}_{arr}^i : Set of flights which are scheduled to arrive at the start of interval i ;

I^f : Set of intervals occupied by flight f , $I^f \subseteq I$;

\mathcal{K}^f : Set of aircraft available to operate flight f , $\mathcal{K}^f \subseteq \mathcal{K}$;

O^f : Set of aircraft preassigned to operate flight $f \in \overline{\mathcal{F}}_R$, $O^f \subseteq \mathcal{K}^f$.

Parameters

DR_{kc}^f : Deterioration for the operation of flight f for check c and aircraft k .

Variables

a_k^f : 1, if the flight f is (re)assigned to aircraft k ; 0, otherwise.

\overline{B}_r : The maximum capacity for resource r .

v_{kc}^f : 1, if the regulation for check c is being violated before flight f for aircraft k ; 0, otherwise.

3.4.3. Formulation

Model 3.2. Interval MMILP formulation for AMS with TA.

$$\min \sum_f \sum_{k \in \mathcal{K}^f} \sum_c v_{kc}^f \quad (18)$$

$$\min \sum_r \overline{B}_r \quad (19)$$

$$\min \sum_f \sum_{k \in \mathcal{K}^f \setminus O^f} a_k^f \quad (20)$$

$$\min \sum_k \sum_c \sum_i (et^i - st^i) m_{kc}^i \quad (21)$$

$$\min \sum_i \sum_r (et^i - st^i) B_r^i \quad (22)$$

$$\min \sum_k \sum_c w_{kc}^{\sup\{I_k\}} \quad (23)$$

Subject to

Preassigned Flights

$$a_k^f = 1 \quad \forall f \in \overline{\mathcal{F}}_R, k \in \mathcal{O}^f; \quad (24)$$

$$\sum_c v_{kc}^f = 0 \quad \forall f \in \overline{\mathcal{F}}_R, k \in \mathcal{O}^f; \quad (25)$$

Maintenance Requirement

$$w_{kc}^i \geq w_{kc}^{i-1} + \sum_{f \in \mathcal{F}_{arr}^i} (DR_{kc}^f a_k^f - v_{kc}^f) - m_{kc}^i \quad \forall k, c, i \neq \inf\{I\}; \quad (26)$$

$$w_{kc}^{\sup\{MOP_k^j\}} \leq \sum_{f \in \mathcal{F}_{dep}^{\sup\{MOP_k^j\}}} DR_{kc}^f a_k^f \quad \forall k, c, j; \quad (27)$$

Reassignment

$$m_{kc}^{i+1} \leq 1 - \sum_{f \in \mathcal{F}_{dep}^i} a_k^f \quad \forall k, c, i \neq \sup\{I\}; \quad (28)$$

$$\sum_k a_k^f = 1 \quad \forall f; \quad (29)$$

Maintenance

Constraints 6, 9 and 10

Transitivity Constraints

Constraints 11 – 13

Resources

Constraints 14

$$B_r^i \leq \overline{B}_r \quad \forall i, r \in R_{Wi}; \quad (30)$$

Variables

$$a_k^f \in \{0, 1\} \quad \forall k, f; \quad (31)$$

$$v_{kc}^f \in \{0, 1\} \quad \forall k, c, f; \quad (32)$$

$$\overline{B}_r \in \mathbb{R}^+ \quad \forall r \quad (33)$$

Constraints 15 – 17

The proposed formulation is a MMILP with six lexicographically ordered objective functions. The functions involved minimise the following objectives (in the order of importance), the number of violations, maximum resource level, number of reassigned flights, number of maintenance interventions, overall resource usage, and total MR. Objective 18, with largest priority, minimises the number of regulation violations. Objective 19 minimises the sum of maximum level for each resource. Objective 20 minimises the number of strictly reassigned flights (i.e. assigned to an aircraft different from the one in the input TA), thus minimising the number of changes in the airlines' preferred TA. Objective 21 minimises the number of maintenance interventions weighted with the duration of intervals. Objective 22 minimises the resource level per interval, again, weighted with the duration of intervals. Finally, as in the previous model, with least priority, objective 23, minimises the total MR at the last interval.

Constraints 24 ensure that if flight f is preassigned to aircraft $k \in \mathcal{O}^f$, then $a_k^f = 1$. Similarly, constraints 25 ensure that if flight f is preassigned to aircraft $k \in \mathcal{O}^f$, then a violation cannot occur $v_{kc}^f = 0$ for any check c .

The MR constraints are a simple extension of those in Model 3.1. Constraints 26 enforce a recurrence relation where the current MR, w_{kc}^i , is updated using the previous MR, w_{kc}^{i-1} , plus a deterioration term if the aircraft has been assigned the flights prior to the interval under consideration, or drops to 0 if either a violation or maintenance occurs. In the case when flight f is reassignable, i.e. $f \in \mathcal{F}_R$, and if flight f is not reassigned to aircraft k , $a_k^f = 0$, then the MR is only updated if a violation or maintenance occurs. On the other hand, in the case of either, flight f being reassigned

to aircraft k , or, flight f being preassigned to aircraft k ($f \in \overline{\mathcal{F}}_R$ and $k \in O^f$), we have, $a_k^f = 1$. Thus, by constraints 25 and 28, $v_{kc}^f = 0$ and $m_{kc}^i = 0$, thus, the MR is deteriorated according to the appropriate deterioration, DR_{kc}^f . Therefore, the recurrence relation is enforced.

Constraints 27 ensure that the MR remains feasible for the operation of any of the flights that depart the end of interval i . That is, for a given aircraft k , check type c and MOP j , the MR at the end of the MOP, $w_{kc}^{\sup\{MOP_k^j\}}$, should remain feasible to operate whichever flights are assigned to aircraft k departing at the end of the MOP. Such flights are contained in $\mathcal{F}_{dep}^{\sup\{MOP_k^j\}}$.

As for the reassignment, constraints 28 ensures that maintenance is not performed if an aircraft departs. More specifically, if an aircraft k is reassigned and due to depart on a flight after interval $i \neq \sup\{I\}$, then, at interval $i + 1$ (after the aircraft has departed) maintenance cannot be performed, hence, $m_{kc}^{i+1} = 0$. Constraints 29 ensure that all flights have exactly one aircraft assigned to them.

Constraints 30 establish the value for the maximum resource level for each resource. The remaining constraints, (maintenance, transitivity and resource constraints) as well as variable definitions, can be borrowed from Model 3.1. The last constraints 31–33, define the domains of the extra variables.

4. Solution Methodology

The solution approach chosen only requires flight schedules, and resource capacities and demands for maintenance services. To improve the efficiency of the solutions, after a preprocessing routine, we implement an iterative algorithm. The algorithm is displayed in Figure 4 and pseudocode presented in Algorithm 1. The algorithm consists of two stages, *conflicting period selection* and *interval splitting*. The conflicting period selection stage involves selecting the sets of reassignable and preassigned flights, and resolving. This stage terminates when all the violations are removed or when the size of the conflicting periods cannot be increased any further. During subsequent iterations, the interval splitting stage identifies intervals where maintenance occurs, splits them and resolves the problem. Splitting time intervals allows the model to assign more maintenance to the existing schedule since it makes time intervals more granular.

4.1. Preprocessing Routine

Flight schedules are crucial for the model as they give the initial TA and accurate FH in order to update the MR throughout the planning horizon. We obtained flight schedules from [Flightradar24 AB \(2018\)](#) using `pyflightdata`, the Python module ([Allamraju, 2014](#)). We gathered data globally for an extended period. Following the data gathering stage, we preprocess the flight schedule data. Preprocessing involves filtering schedules through the nearest airport to the maintenance workshops under consideration. After this, we identify airlines and aircraft types of interest so we can track and update the aircraft's FH appropriately. This gives a reduced network with accurate FH for each aircraft. For eight maintenance workshops over a 30-day planning horizon (between dates 14/11/16 and 15/12/16), prior to preprocessing, we have 23927 flights and 1643 aircraft of two types (Airbus A320 and Fokker 100). Then, we proceed to identify MOPs i.e. turnaround times sufficient to perform at least the shortest maintenance type. We choose a turnaround time of at least 5 hours to allow for at least a short maintenance intervention. Using aircraft MOPs, we generate intervals by identifying all start and end times of the MOPs and storing them in an ordered set.

As part of this stage, we pre-determine which variables are not required in the model, hence, for efficiency, they are not created. Specifically, all the variables involved in constraints 5 and 6 in Model 3.1, and constraints 24, 25, and 28 in Model 3.2, have known values and it is unnecessary to create them. Therefore, in practice, since the reassignment variables that involve preassigned flights are not present in Model 3.2, when MR constraints involve preassigned flights, we simply use MR constraints from Model 3.1. For this reason, the number of reassignment variables is determined by selecting the sets of reassignable and preassigned flights.

4.2. Solution Procedure

To ensure efficiency and solution accuracy, we propose an iterative solution procedure which has two stages, conflicting period selection, and interval splitting. The aim of the conflicting period selection stage is to, by using the identification of regulation violations, select the number of reassignment variables. In each iteration, if the regulation violations involved have not been removed, we increase the size of the conflicting period (which determines the

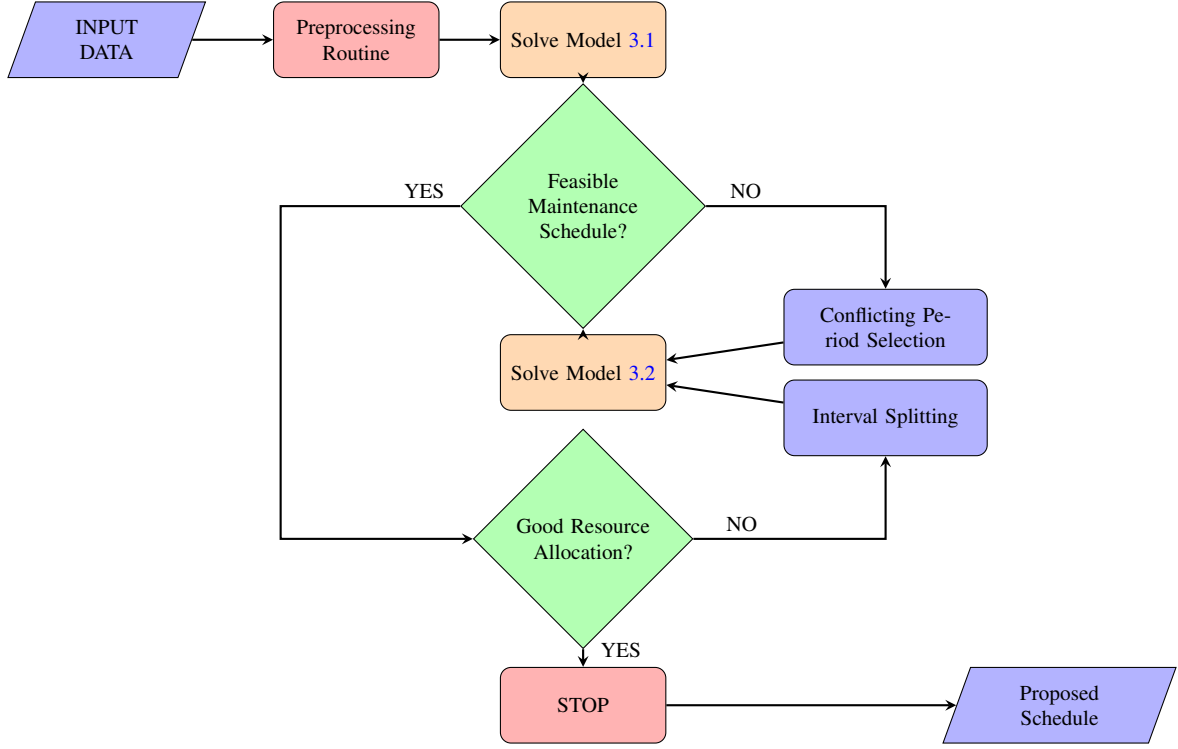


Figure 4: Flow chart outlining the process of the iterative algorithm.

number of reassignment variables) and resolve Model 3.2. In the interval splitting stage, once all infeasibilities have been removed or all reassignment variables have been introduced, we split intervals where maintenance takes place and resolve Model 3.2. This ensures flexibility as it allows the additional generated intervals to be allocated to different aircraft.

4.2.1. Conflicting Period Selection

An illustration of the conflicting period selection stage is given in Figure 5. For a certain violation, say at interval i , with $\{k_1, k_2, k_3, \dots, k_K\} \in \mathcal{K}^i$, we select the initial conflicting period as shown in red with $j_{(1)}^i$ as the upper bound, as the next interval where $\mathcal{K}^i \subseteq \mathcal{K}^{j_{(1)}^i}$. We classify flights either in the reassignable subset (\mathcal{F}_R), if they are involved in the conflict, or in the preassigned subset ($\overline{\mathcal{F}}_R$), otherwise. This regulates the number of reassignment variables, and keeps the solution process of Model 3.2 efficient. If in the new solution, interval i still has a violation, we increase the size of the conflicting period to reach $j_{(2)}^i$, the next interval where $\mathcal{K}^i \subseteq \mathcal{K}^{j_{(2)}^i}$. Again, we update the sets of reassignable and preassigned flights and resolve the problem. If the violation has not been removed, we move on to the next interval with matching aircraft. We continue this process until the violation is removed or the end of the planning horizon is reached, in which case, all the reassignment variables, corresponding to violation i , would have been introduced.

In order to identify the conflicting period, we have to identify where violations occur. For the first iteration, the set V that contain all intervals for which at least one violation occurs is given by,

$$V = \left\{ i : i \in \mathcal{I}, \sum_k \sum_c v_{kc}^i \geq 1 \right\},$$

thereafter, it is given by,

$$V = \{i : i \in \mathcal{I}^f, f \in F\},$$

where,

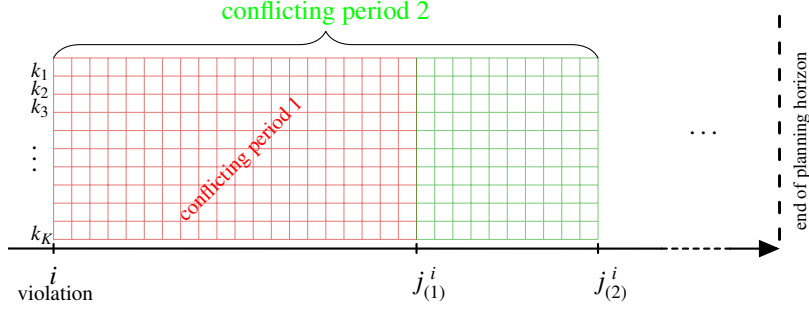


Figure 5: Conflicting period selection.

$$F = \left\{ f : f \in \mathcal{F}, \sum_k \sum_c v_{kc}^f \geq 1 \right\}.$$

For every interval with a violation, $i \in V$, we can identify the set of aircraft involved, \mathcal{K}^i . Then, we can find the interval where the aircraft in \mathcal{K}^i will meet again, say \mathcal{K}^j , with $\mathcal{K}^i \subseteq \mathcal{K}^j$, where $j > i$. Hence, the reassignable flights, \mathcal{F}_R , are those that occur between intervals i and j , the conflicting period, and for $k \in \mathcal{K}^i$. All the flights, outside of the conflicting period are preassigned, thus, $\overline{\mathcal{F}}_R = \mathcal{F} \setminus \mathcal{F}_R$. More generally, the set of intervals J^i , which contain the set of aircraft \mathcal{K}^i , can be written as,

$$J^i = \{j : j \in \mathcal{I}, j > i, \mathcal{K}^i \subseteq \mathcal{K}^j\}. \quad (34)$$

With this, we can write the first conflicting period as

$$CP(1, i) = \{i' : i' \in \mathcal{I}, i \leq i' \leq j_{(1)}^i, \exists k \in \mathcal{K}^{i'} \wedge k \in \mathcal{K}^i\}, \quad (35)$$

where $j_{(1)}^i$ is the first element in J^i . We use this to update the sets of reassignable and preassigned flights appropriately. The set of reassignable flights is given by,

$$\mathcal{F}_R = \{f : \inf\{\mathcal{I}^f\} \geq i, \sup\{\mathcal{I}^f\} \leq j_{(1)}^i\},$$

and $\mathcal{K}^f = \mathcal{K}^i$ for $f \in \mathcal{F}_R$. The set of preassigned flights can be found using the updated set of reassignable flights, while O^f , the set of aircraft preassigned to operate flight $f \in \overline{\mathcal{F}}_R$ is given by the initial TA. Additionally, we need to update the aircraft present at the intervals in the conflicting period. Thus, we set

$$\mathcal{K}^{i'} = \mathcal{K}^{i'} \cup \mathcal{K}^i \text{ for } i' \in CP(1, i).$$

By solving Model 3.2 after these updates, we can reassign the flights to any of the aircraft involved in the conflict. In the next iterations, if the violation is removed, then without loss of generality, we may assume that the remaining schedule can remain unchanged. However, if solving Model 3.2 has not led to the elimination of the violation, we expand the conflicting period and resolve. For this, we use $CP(m, i)$, for $m = 2, \dots, |J^i|$, which uses the m -th element of J^i , $j_{(m)}^i$. Similarly, to update the set of reassignable flights, we set,

$$\mathcal{F}_R = \{f : \inf\{\mathcal{I}^f\} \geq i, \sup\{\mathcal{I}^f\} \leq j_{(m)}^i\},$$

with, $\mathcal{K}^f = \mathcal{K}^i$ for $f \in \mathcal{F}_R$, and,

$$\mathcal{K}^{i'} = \mathcal{K}^{i'} \cup \mathcal{K}^i \text{ for } i' \in CP(m, i).$$

With the set of preassigned flights and aircraft being updated as previously, using the new set of reassignable flights. We do this for every interval with a violation $i \in V$, until either the violation is eliminated or the end of the planning horizon is reached.

4.2.2. Interval Splitting

The interval splitting stage favours the redistribution of resources by using a more granular timeline with each iteration. Resources are occupied for the duration of the interval if maintenance is being performed. When long intervals occur, given the non-preemption assumption, resources can be held even after the maintenance has been finalised (exceeding the minimum maintenance duration). Therefore, we consider the effect of splitting intervals, using different criteria, and resolving the problem. For example, in Figure 6, given the intervals, in the first iteration we have a MOP of 9 hours. Suppose that the maintenance scheduled for this MOP only takes 8 hours. So, in the first iteration, the last hour of the last interval is being wastefully allocated. Splitting, therefore, allows for the resources to be allocated to different aircraft. By splitting in half, in iteration 3, we see that the last hour is no longer being held.

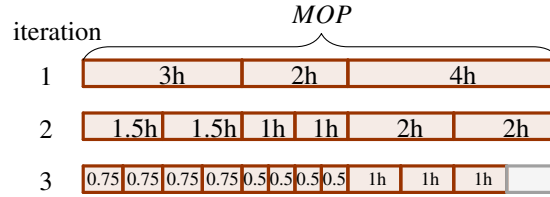


Figure 6: Interval splitting stage for a 9 hour MOP using binary segmentation for three iterations.

Let us define three simple splitting methods. Suppose, after solving the problem, we have $m_{kc}^i = 1$, so we split interval i using the following,

1. Binary segmentation (split in half);
2. Golden ratio (split by Golden ratio);
3. Minimum cut (split that allows at least the shortest type of maintenance).

Methods 1 and 2 are well-known and are regularly employed in search algorithms (Nocedal and Wright, 2006). Method 3 splits intervals that allows the shortest type of maintenance. Going back to the example in Figure 6, recall that the duration of the maintenance is 8 hours, splitting using Method 3 would produce a single split at the 8th hour in the second iteration.

To compare the splitting methods and identify good solutions, we compute an accuracy measure representing the usage of MOPs. The accuracy measure can be expressed as the ratio of the minimum time required for a check over the actual time scheduled for the check. That is,

$$A = \frac{\sum_c \sum_k \sum_j \Delta_{t(k),c}}{\sum_c \sum_k \sum_j \sum_{i \in MOP_k^j} (e^t - s^t) m_{kc}^i}. \quad (36)$$

Given constraints 10, which specify that maintenance scheduled should be at least of the minimum required duration; we have that $A \leq 1$. Therefore, a schedule that has an accuracy value close to 1, is one that does not schedule more maintenance than strictly required, and is, thus, efficient.

4.2.3. Algorithm

As outlined in Figure 4 and in the pseudocode of Algorithm 1, the algorithm requires some inputs. Specifically, a set of intervals, set of aircraft, FH for each aircraft, maintenance regulations and durations, and, resource capacities and demands. Once these are provided, in the first iteration, Model 3.1 is solved. If there are violations; i.e. there is no feasible maintenance schedule or, equivalently, $V \neq \emptyset$; we implement the conflicting period selection stage and resolve Model 3.2 until the violations have been removed. After this, we iterate the interval splitting stage and resolve Model 3.2 until either, the resource allocation is good enough, equivalently, $A \geq 1 - \varepsilon$ (where ε is the tolerance), or, no more intervals are added.

Algorithm 1 Solution procedure with conflicting period selection and interval splitting.

```

1: Initialisations
2: iter, A = 0                                ▶ Iteration counter and accuracy measure
3:  $\varepsilon = 0.01$                             ▶ Set Tolerance
4: maxiter = 50                                ▶ Maximum number of iterations
5: Intervals =  $\mathcal{I}$                             ▶ Interval set
6: V, CP, oldIntervals = []                    ▶ Empty array for violations, conflicting periods, and old intervals
7: while iter < maxiter and A < 1 -  $\varepsilon$  do
8:   if iter = 0 then
9:     Solve Model 3.1                            ▶ For the first iteration
10:  else
11:    Solve Model 3.2                            ▶ For iterations  $\geq 1$ 
12:  end if
13:  Update V and A                                ▶ Update violations and accuracy measure using new solution
14:  if V is not empty then
15:    SELECTCP(V)                                ▶ If violations have not been removed
16:    else
17:      which =  $\{i: i \in \mathcal{I}, \sum_k \sum_c m_{kc}^i \geq 1\}$     ▶ Identify intervals with maintenance scheduled
18:      oldIntervals = Intervals
19:      SPLITINTERVALS(which, Intervals)        ▶ Call function to split intervals
20:      if |Intervals| = |oldIntervals| then    ▶ No intervals have been added
21:        Break                                    ▶ Stop the algorithm
22:      end if
23:    end if
24:    Increment iter
25:  end while
26:
27: function SELECTCP(V)                            ▶ Updates the conflicting periods for all violations
28:   for i in V do
29:     Compute  $J^i$ 
30:     if i not in CP then
31:       CP = CP  $\cup$  CP(1, i)                    ▶ If the violation is new
32:     else
33:       for m in 2  $\rightarrow$  | $J^i$ | do
34:         if CP(m-1, i) in CP then
35:           CP = CP  $\setminus$  CP(m-1, i)        ▶ Update CP with first conflicting period
36:           CP = CP  $\cup$  CP(m, i)              ▶ If violation has occurred at a previous iteration
37:         end if
38:       end for
39:     end if
40:   end for
41:   Intervals = Intervals  $\cup$  CP                ▶ Iterate through elements in  $J_i$ 
42: end function                                ▶ Find already used CP
43:
44:
45: function SPLITINTERVALS(which, Intervals)    ▶ Remove it from conflicting periods
46:   for i in which do
47:     Using Method 1, 2, or 3; split i into  $i_1$  and  $i_2$ 
48:     Intervals = Intervals  $\setminus$  {i}  $\cup$  { $i_1, i_2$ }    ▶ Update Intervals
49:   end for
50: end function

```

5. Model Application and Computational Tests

We tested the iterative algorithm using flight schedules obtained for the maintenance workshops under consideration over the 30-day period selected. The results shown in this section use preprocessed flight data between dates 14/11/16 and 15/12/16 (see Section 4.1 for more information on the data gathering and preprocessing). The iterative algorithm was written in Python, using [Gurobi Optimization version 8.0 \(2018\)](#) to solve the models. Two computational studies are presented, one for the single workshop case (five workshops treated independently) and one for the multi-workshop case (up to eight workshops treated simultaneously).

As for the experimental set-up, we restrict the interval splitting stage of the algorithm such that the resulting intervals are at least 5 seconds long. Solution times for each iteration are limited to 500 seconds for each objective. The parameters used for the computational tests include the standard duration and frequencies for the regulated medium-term maintenance checks, as mentioned in Section 1. Specifically, according to Table 1, for the two types of aircraft under consideration, the values of regulation parameter, $L_{t(k),c}$, is given in Table 3. Here, we see the type of aircraft (Airbus A320 and Fokker 100), check type ($c = 1$ or 2) and the corresponding maintenance regulation parameter values, $L_{t(k),c}$. Using these values, $c = 1$ corresponds to an A check and $c = 2$ corresponds to a C check. Resource demand and capacity bounds, shown in Table 4, vary for four different types of renewable resources. A realistic interpretation of resources is as follows, r_1 – number of hangar bays, r_2 – certified technicians, r_3, r_4 – different types of specialised tools. The value for the tolerance, ε , which determines the required accuracy level as $A \geq 1 - \varepsilon$, is set to be 0.01. The initial MR variable is sampled from a Uniform distribution as follows,

$$w_{kc}^{\inf\{I_k\}} \sim Unif(0, 0.3) \forall k, c.$$

Sensitivity analysis around the chosen value produces distinct maintenance schedules and resource profiles but does not affect the computational performance of the algorithm. For instance, higher values lead to more maintenance being scheduled towards the start of the planning horizon; conversely, lower values lead to more maintenance being scheduled towards the end of the planning horizon.

Table 3: Maintenance regulation parameter values for two check types (1 and 2) and two aircraft types (A320 and F100).

$t(k)$	$c = 1$	$c = 2$
Airbus A320	$L_{t(k),c} = 600\text{FH}$	$L_{t(k),c} = 6000\text{FH}$
Fokker 100	$L_{t(k),c} = 500\text{FH}$	$L_{t(k),c} = 5000\text{FH}$

Table 4: Sample resource demands and limit capacities for four types of resources (r_i for $i = 1, 2, 3, 4$) and two maintenance types (1 and 2).

r	b_{r1}	b_{r2}	\bar{B}_r
r_1	1	1	25
r_2	3	5	25
r_3	2	3	25
r_4	1	2	25

In addition, the formulations are solved using lexicographic ordered objectives with priorities as suggested by airline practitioners. In the order of importance, the objectives which are minimised, namely,

1. the number of violations (objective 18),
2. the maximum resource level (objective 19),
3. the number of strictly reassigned flights (objective 20),
4. the weighted number of maintenance interventions (objective 21),
5. the total MR at the end of the planning horizon (objective 23).

Given that we are seeking for a maintenance feasible schedule, assigning the number of violations, in any but the highest priority level leads to infeasible maintenance schedules. Apart from this, as tests revealed, the order of the other objectives does not affect the integrity of the solutions or the computational times.

5.1. Single Workshop Case

In the case when maintenance workshops can be treated independently, we can solve the problem for each workshop individually. This situation can occur, for example, due to the geographical location of the workshops or upon a particular airlines' request or restrictions. In this case, we implement the algorithm in parallel for each maintenance workshop under consideration. It is worth noting that due to the preprocessing routine, which leads to considering only flights with large turnaround times and just two specific aircraft types, the number of flights and aircraft are significantly reduced compared those seen in ordinary operations; for more information see Section 4.1. The details for the five workshops, are as follows,

Atlanta Hartsfield-Jackson International Airport: with 1048 flights and 115 aircraft, produces 389 intervals;

Bangkok Suvarnabhumi Airport: with 3869 flights and 49 aircraft, produces 843 intervals;

Cairo International Airport: with 781 flights and 34 aircraft, produces 279 intervals;

Dubai International Airport: with 223 flights and 11 aircraft, produces 63 intervals;

Tokyo Haneda International Airport: with 978 flights and 10 aircraft, produces 162 intervals.

In all cases under consideration, after a single iteration of the conflicting period selection stage, violations are removed and the interval splitting stage begins. In order to compare the three splitting methods, we study four different aspects, namely, the final number of intervals, total run times, accuracy measure, and objective function value. Table 5 breaks down the objective function for the largest workshop, while the results for the workshops under consideration are shown in Table 6. Specifically, Table 6 shows the accuracy measure plots per iteration, final number of intervals, and total run times. In the accuracy measure plots, as shown in the legend, Method 1 is represented with solid lines, Method 2 with dashed lines, and Method 3 with dot-dashed lines.

The trend with the number of intervals per iteration is increasing for all methods, which is expected. The final number of intervals, as shown in Table 6, is the number of intervals at the last iteration. Method 3 offers the least final number of intervals throughout, which is reflected in its solution times. Method 1 reveals a higher number of intervals than Method 2.

The run times for the largest workshop, Bangkok, using Method 3 takes 72 minutes to terminate the algorithm. For the remaining, small to medium-sized workshops, it takes between 0.36 seconds to 4 minutes to reach a good solution. Computational times for Method 1 and Method 2 (both significantly larger than Method 3) with the latter showing lower solution times.

The accuracy measure per iteration appears as a plot in the second column in Table 6. As can be seen, the accuracy measure evolves differently depending on the workshop. It takes a varying number of iterations across workshops for the algorithm to terminate. In all cases, more clear for Atlanta and Dubai, the required level of accuracy (0.99) is not reached; the algorithm terminates due to no more intervals being created. For the cases of Tokyo and Dubai, using Method 3, terminates the algorithm in very few iterations; whereas the rest take slightly more. Between Methods 1 and 2, both provide very similar quality solutions.

A breakdown of the different components of the objective function value for the Bangkok workshop is given in Table 5. The table shows the different values for each objective per iteration and for the corresponding splitting method. In each iteration, the best method is shown in green. This is determined by comparing objectives in decreasing order of priority until differing objectives are found, and one method presents an objective value lower than the rest. Note that objectives 18 (number of violations) and 20 (number of reassigned flights) remain constant. Objectives 21 (weighted number of maintenance interventions) and 22 (weighted overall resource usage) decrease, while objectives 19 (maximum resource level) and 23 (total MR at the end of the planning horizon) show some fluctuation. The decrease in the fourth and fifth priority objectives is due to them being the only ones weighted with the duration of the intervals, and, therefore, are the only ones that are decreasing as the duration of maintenance interventions also decrease. The first iteration is the starting point where no splitting has occurred, hence, objectives have the same values throughout the three splitting methods. In the second iteration, Method 2 is better than Method 3 as it presents the same first three priority objectives (Obj. 18 - 20) and a lower value in the fourth priority objective (Obj. 21). Iterations thereafter show that Method 3 dominates with a lower second priority objective (Obj. 19). Additionally, Method 3 provides healthier fleet overall, as suggested by the consistently lower value of objective 23.

Due to its good solution times, lowest objective values, and overall higher accuracy measure we can claim that, for the single workshop case, the interval splitting stage performs better using Method 3. Furthermore, it is worth noting that the aircraft are, also, kept in a healthier state at the end of the planning horizon. For this reason, we present more detailed results using Method 3 for the largest workshop (Bangkok).

Figure 7 shows two resource profiles (first and last iteration) for resource r_1 at the Bangkok workshop. From Figure 7a to Figure 7b, we see that the resource profiles become considerably less populated and thinner. This means that the resource usage is more efficient. Particularly, by the last iteration, the solution tends to have slightly higher resource levels to avoid performing maintenance during busy flight periods. This leads to most maintenance occurring during night and early morning shifts, as one would expect.

Since terminating the algorithm when $A \geq 1 - \varepsilon$, or when no more intervals can be added, does not guarantee an optimal solution we conducted some further testing. For the Atlanta workshop, we compare our solution (using Method 3) to one obtained with a traditional discretisation method. We discretised time intervals using a varying time step, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 minutes, and solved the problem once with a 5 hour time restriction per objective. Figure 8 shows, on the left-hand y-axis, the percentage error when comparing the optimal objectives 21 (weighted number of maintenance interventions) obtained using our method with the traditional discretisation for different time steps. The right-hand y-axis, with a grey dashed line, shows the computational times when using the traditional discretisation for different step sizes.

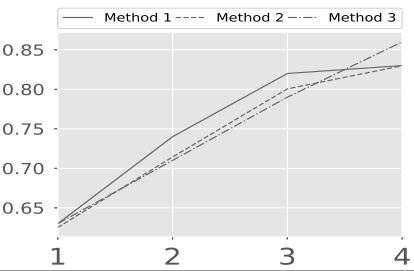
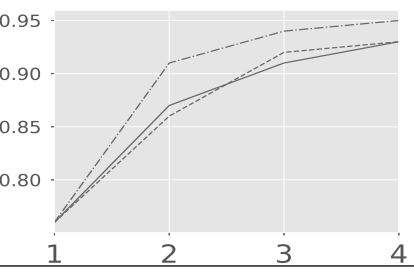
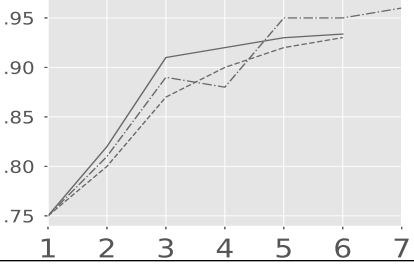
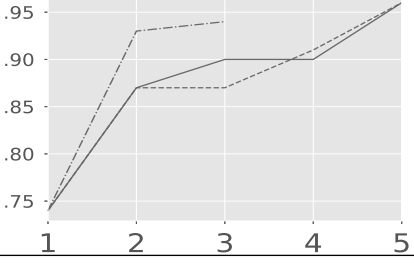
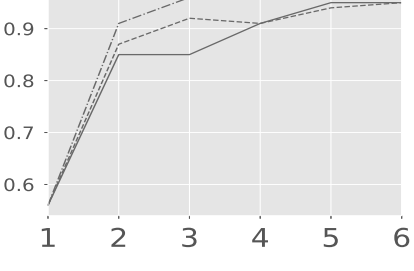
The percentage error for objectives 21, and 22, are shown using a solid red, and blue lines respectively. The corresponding error for the lower bounds are shown in the same colours but with dot-dashed lines. As with previous cases, objectives 18–20 remain constant, so they have not been included in the figure. It can be observed that the percentage error increases as the time step decreases. This means that the solution obtained with the traditional discretisation is improving, with respect to our optimal solution, as the step size is reduced. Our solution provides better solutions for step sizes of 8 minutes and above (hence the negative percentage error values). The optimal solution for a time step of 4 minutes (the smallest available as 2 and 3 minutes timed-out) shows only 0.6% improvement for both objectives. While for the lower bounds for the smallest time step (2 minutes) show 6.6% improvement in objective 21, and 8% in objective 22. Nevertheless, our solution gives a value 20% lower over all step sizes for the second priority objective (objective 19). Thus, given the priority of the objectives, Method 3 dominates all the solutions studied produced using traditional discretisation, thus, producing near optimal solutions.

The dashed line in Figure 8, plotted against the right-hand y-axis, reveals that implementing a traditional time discretisation comes at a huge computational cost. Specifically, the discretisation with the smallest time step (2 minutes) is around 188 times slower than our solution. The reason why our method is significantly more efficient is because it only makes time intervals more granular when it is required and where the solution is more sensitive.

Table 5: Objective function breakdown for the Bangkok workshop. The best method in each iteration is highlighted in green.

Iteration	Method / Obj.	Obj. 18	Obj. 19	Obj. 20	Obj. 21	Obj. 22	Obj. 23
1	Method 1	0	70	3	305	2184	54
	Method 2	0	70	3	305	2184	54
	Method 3	0	70	3	305	2184	54
2	Method 1	0	70	3	216	1559	62
	Method 2	0	70	3	209	1509	52
	Method 3	0	70	3	210	1523	60
3	Method 1	0	77	3	179	1304	66
	Method 2	0	77	3	179	1303	60
	Method 3	0	70	3	173	1258	60
4	Method 1	0	77	3	169	1231	71
	Method 2	0	77	3	169	1234	101
	Method 3	0	70	3	169	1233	55

Table 6: Comparison across 5 workshops during the interval splitting stage for the three different splitting methods.

Station	Accuracy Measure (per iteration)	Features	Splitting Methods		
			Method 1	Method 2	Method 3
Atlanta			Method 1	Method 2	Method 3
			Final # intervals	3343	3187
		Total CPU time (min)	60	52	4
Bangkok			Method 1	Method 2	Method 3
			Final # intervals	2713	2653
		Total CPU time (min)	105	95	72
Cairo			Method 1	Method 2	Method 3
			Final # intervals	3431	3117
		Total CPU time (min)	68	47	5.5
Dubai			Method 1	Method 2	Method 3
			Final # intervals	473	457
		Total CPU time (s)	6.6	4.4	0.36
Tokyo			Method 1	Method 2	Method 3
			Final # intervals	845	771
		Total CPU time (min)	234	72	3.3

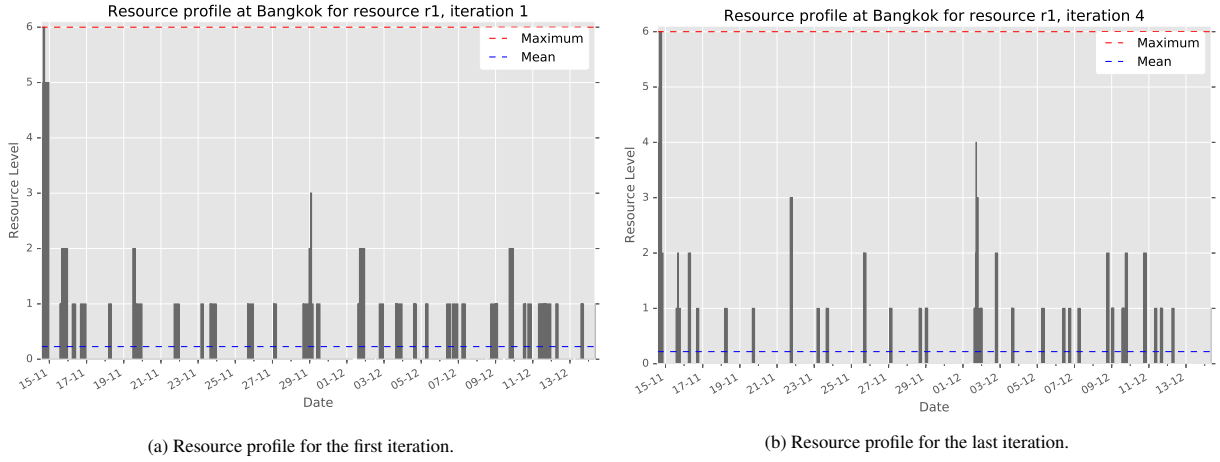


Figure 7: Resource profiles for the Bangkok workshop for the first and last (4th) iterations of the interval splitting stage using Method 3.

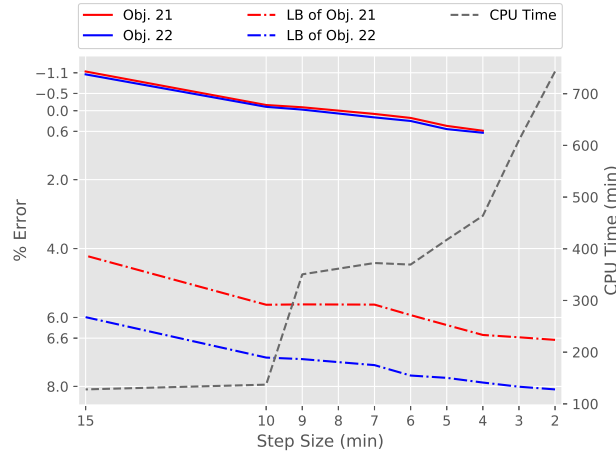


Figure 8: Result comparison (objectives 21, 22) for Method 3 vs traditional discretisation for different time steps.

5.2. Multi-workshop Case

To allow for interdependence between airlines and workshops, we consider several multi-workshop cases. Here, the algorithm solves the problem for all the workshops simultaneously. The first multi-workshop case considers four of the workshops used in the single workshop case. Using Atlanta, Cairo, Dubai and Tokyo, we produce a multi-workshop test set with 3002 flights and 167 aircraft, which produces 900 intervals. The interdependence for this case, due to their close proximity, is between Cairo and Dubai.

After a single iteration of the conflicting period selection, all violations are removed. Hence, the interval splitting stage begins. Table 7 shows, the intervals, computational time and accuracy measure value for each of the splitting methods. In each iteration, the best method is shown in green. As with the single workshop, this is determined by comparing objectives in decreasing order of priority until differing objectives are found, and one method presents an objective value lower than the rest. To avoid repetition, these have not been included due to their resemblance with the results in the single workshop case. The computational times reveal that Method 3 is the fastest (with a total of 1158 intervals and 9 minutes), followed by Method 2 (with a total of 4812 intervals and 58 minutes) and then Method 1 (with a total of 5020 intervals and 62 minutes). The highest accuracy measure, however, is no longer associated with Method 3, but with Method 1.

Table 7: Method comparison for the 4-workshop case.

Iteration	Method	Intervals	CPU time (s)	Accuracy Measure
1	Method 1	900	59	0.72
	Method 2	900	57	0.72
	Method 3	900	61	0.72
2	Method 1	1384	610	0.86
	Method 2	1390	414	0.83
	Method 3	1076	123	0.84
3	Method 1	2556	1252	0.93
	Method 2	2586	1258	0.92
	Method 3	1134	148	0.90
4	Method 1	5020	1777	0.99
	Method 2	4812	1739	0.98
	Method 3	1158	200	0.94

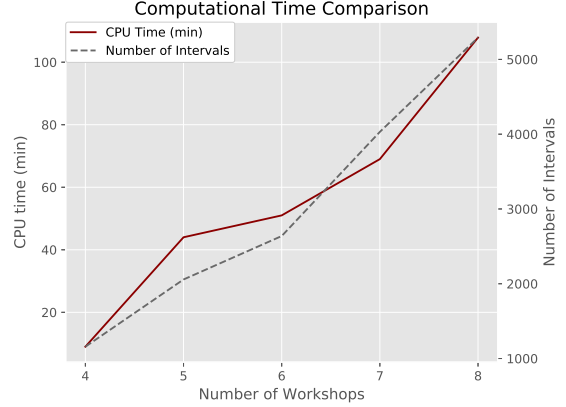


Figure 9: Computational comparison for Method 3 applied to different multi-workshop cases.

Given that it shows lowest solution times and objective values, provides a healthier fleet, and exhibits an acceptable accuracy measure we suggest that, for the multi-workshop case, the interval splitting stage should be implemented using Method 3. However, if more accuracy is sought, one can resort to Method 1 at the expense of higher computational times. When compared with the average accuracy measure obtained individually for the four workshops, Method 1 shows a 6% increase. This suggests that modelling interdependence between workshops, or, inter-airline cooperation, leads to more efficient allocation of resources.

Figure 10 shows the maintenance schedules produced using Method 3 for the four workshops involved in the multi-workshop test set. The x -axis represents the date and the y -axis shows each individual aircraft tail number. Hence, the plot shows when in the planning horizon a specific aircraft is being served, which type of maintenance it is receiving, and where the maintenance is taking place. Note that only one check 2 ($c = 2$ which corresponds to a C check) is scheduled in the busiest workshop, Atlanta, on the first day (14/12). The rest of the maintenance shown are checks $c = 1$ which correspond to A checks.

To study the scalability of the framework, four additional multi-workshop cases were created. They include, a 5 workshop case with 6871 flights and 216 aircraft, a 6 workshop case with 9681 flights and 260 aircraft, a 7 workshop case with 12402 flights and 429 aircraft, and, the largest, an 8 workshop case with 16000 flights and 529 aircraft. Additional to the five workshops considered in the single workshop case, the 8 workshop case considers workshops based in Abu Dhabi International Airport, Beijing Capital International Airport, and, Madrid Barajas Airport. These cases were solved using Method 3 and until the algorithm was terminated; the least accuracy level achieved was of 0.97. The resulting computational comparison, in Figure 9, shows the computational times (red line on the left-hand y -axis) and final number of intervals (grey dashed line on the right-hand y -axis) for the different multi-workshop cases. The evolution of the CPU time and number of intervals seems to increase linearly with the number of workshops; the 8 multi-workshop case finalises with 5292 intervals and in 108 minutes. Therefore, it is possible that the framework is scalable for larger data sets. Clearly, this depends on the size and number of flights introduced each new workshop. Nonetheless, given the size of the largest instance considered, we have managed to demonstrate the computational efficiency and the potential of this framework.



Figure 10: Maintenance schedules produced by using Method 3 for each aircraft and workshop in the first multi-workshop case.

6. Conclusions

We have solved the airline fleet maintenance scheduling problem considering tail assignment. Previous studies that tackle a similar problem have modelled a short-term planning horizon and tend to be computationally expensive even for moderately sized data sets. By using aircraft individual legal remaining flying hours and our interval based formulations, we tackle the problem while providing solutions in reasonable time. We present two multi-objective mixed integer linear programming formulations for this purpose. The first acts as a feasibility check for an input flight schedule and provides an initial set of variables for the tail assignment. If a feasible maintenance schedule was not found, the second formulation employs the location of the regulation violations to formulate a combined maintenance and tail assignment problem. This explores the different options across an aircraft journey and decides on the optimal allocation of flights, maintenance and resources. Additionally, our approach accounts for multiple resources and for generic types of maintenance.

In order to improve solutions, we implement a heuristic algorithm that consists of two stages: conflicting period selection and interval splitting. The conflicting period selection stage increases the size of the tail assignment problem gradually until we are able to produce a feasible maintenance schedule. After this, the interval splitting stage improves resource allocation.

Test results show that the algorithm is efficient, since it can solve large instances in reasonable computational time, for a 30-day planning horizon and provides good quality solutions. We solved a multi-workshop test case with 8 workshops, 16000 flights and 429 aircraft in under 2h. Highlighting the importance of developing new efficient formulations. Solutions present airlines with alternatives to their initial tail assignment during the planning stage. These solutions focus on satisfying maintenance regulations and keeping the aircraft healthy, while remaining commercially viable for the airlines. Results promote inter-airline cooperation, which allows for workshop resource sharing between airlines, since it provides a more efficient resource allocation per workshop.

Some limitations are worth noting. Stopping the iterative algorithm when the accuracy measure is maximal does not guarantee an optimal solution. Clearly, if continued indefinitely, the algorithm would reach the optimal solution, however, applying the stopping criterion gives a good quality solution in reasonable time. We compared the solution obtained using our method to the one obtained using a traditional discretisation with a small time step. This revealed that our solution is better and around 188 times faster, however, the theoretical proof remains to be done. Additionally, we have not considered the complications that may arise from some of the long-term maintenance types. In some cases, for example with life limited replacements, the inclusion of inventory control for spares and those being fixed would be paramount. Further work includes the implementation of a rolling horizon and the application of clustering for maintenance workshops with intersecting flights. Due to the lexicographic optimisation approach, the optimal solution(s) of the higher order objectives determine the optimal values of the lower order objectives. The solution space of the proposed formulation can be further expanded by relaxing the optimal value of the first priority objective, i.e. allowing for tolerance. An alternative way to model the problem is to use a Goal Programming formulation and seek to minimise the weighted sum of the deviation from the optimum value of each objective. It is apparent that, the preferences of the decision maker(s) in terms of preference ordering (lexicographic optimisation) or weights (Goal Programming) will drive the outcome of the optimisation process. Future research may involve further experimentation of the decision maker(s) preferences in terms of the preferences of the decision maker(s).

Acknowledgements

We thank Richard Standing, from R² Data Labs (Rolls-Royce), for his guidance and suggestions. We would also like to thank the reviewers for their helpful comments that shaped and improved this paper. We gratefully acknowledge the support of the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training and Rolls-Royce.

References

- Ackert, S., 2011. The Relationship between an Aircraft's Value and its Maintenance Status. Tech. rep., Aircraft Monitor, accessed: 07/01/2020. http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/the_relationship_between_an_aicrafts_value_its_maintenance_status___v1.pdf
- Aircraft Analysis & Fleet Planning, 2005. Can the 787 & A350 transform the economics of long-haul services? Tech. rep., Aircraft commerce, accessed: 07/01/2020. <https://www.aircraft-commerce.com/articles/articles-by-subject/>

- Allamraju, H., 2014. Flightradar24 Unofficial API: pyflightdata. Accessed: 07/01/2020. <http://pyflightdata.readthedocs.io/en/latest/about.html>
- 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. <http://dx.doi.org/10.1287/trsc.32.3.208>.
- Brucker, P., Knust, S., 2012. *Complex Scheduling*, 2nd Edition. GOR-Publications. Springer, Berlin, Heidelberg. <http://dx.doi.org/10.1007/3-540-29546-1>.
- Clarke, L., Johnson, E., Nemhauser, G., Zhu, Z., 1997. The aircraft rotation problem. *Annals of Operations Research* 69 (0), 33–46. <http://dx.doi.org/10.1023/A:1018945415148>.
- Cohn, A. M., Barnhart, C., 2003. Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions. *Operations Research* 51 (3), 387–396. <http://dx.doi.org/10.1287/opre.51.3.387.14759>.
- Cook, A., Tanner, G., 2008. Innovative Cooperative Actions of R&D in EUROCONTROL Programme CARE INO III: Dynamic Cost Indexing: Aircraft maintenance – marginal delay costs. Tech. rep., Transport Studies Group, University of Westminster, accessed: 07/01/2020. https://www.eurocontrol.int/eec/gallery/content/public/documents/projects/CARE/CARE_INO_III/DCI_TDD9-0_Airline_maintenance_marginal_delay_costs.pdf
- Cordeau, J.-F., Stojković, G., Soumis, F., Desrosiers, J., 2001. Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling. *Transportation Science* 35 (4), 375–388. <http://dx.doi.org/10.1287/trsc.35.4.375.10432>.
- Department for BIS, 2016. UK Aerospace Maintenance, Repair, Overhaul & Logistics Industry Analysis. Accessed: 07/01/2020. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/502588/bis-16-132-uk-mrol-analysis.pdf
- Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M. M., Soumis, F., 1997. Daily aircraft routing and scheduling. *Management Science* 43 (6), 841–855. <http://dx.doi.org/10.1287/mnsc.43.6.841>.
- Feo, T. A., Bard, J. F., 1989. Flight scheduling and maintenance base planning. *Management Science* 35 (12), 1415–1432. <http://dx.doi.org/10.1287/mnsc.35.12.1415>.
- Flightradar24 AB, 2018. Flightradar24. www.flightradar24.com, accessed: 07/01/2020.
- Gopalan, R., Talluri, K. T., 1998. The Aircraft Maintenance Routing Problem. *Operations Research* 46 (2), 260–271. <http://dx.doi.org/10.1287/opre.46.2.260>.
- Gurobi Optimization version 8.0, I., 2018. Gurobi Optimizer Reference Manual. Accessed: 07/01/2020. <https://www.gurobi.com/documentation/8.0/refman/index.html>
- Habibi, F., Barzinpour, F., Sadjadi, S., 2018. Resource-Constrained Project Scheduling Problem: Review of Past and Recent Developments. *Journal of Project Management* 3 (2), 55–88. <http://dx.doi.org/10.5267/j.jpm.2018.1.005>.
- Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., Sigismondi, G., 1993. The fleet assignment problem: solving a large-scale integer program. *Mathematical Programming* 70 (1-3), 211–232. <http://dx.doi.org/10.1007/BF01585938>.
- Haouari, M., Shao, S., Serali, H. D., 2013. A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem. *Transportation Science* 47 (4), 508–525. <http://dx.doi.org/10.1287/trsc.1120.0433>.
- Hicks, R., Madrid, R., Milligan, C., Pruneau, R., Kanaley, M., Dumas, Y., Lacroix, B., Desrosiers, J., Soumis, F., 2005. Bombardier Flexjet Significantly Improves Its Fractional Aircraft Ownership Operations. *Interfaces* 35 (1), 49–60. <http://dx.doi.org/10.1287/inte.1040.0113>.
- Khaled, O., Minoux, M., Mousseau, V., Michel, S., Ceugniet, X., 2018. A Compact Optimization Model for the Tail Assignment Problem. *European Journal of Operational Research* 264 (2), 548–557. <http://dx.doi.org/10.1016/j.ejor.2017.06.045>.
- Koné, O., Artigues, C., Lopez, P., Mongeau, M., 2011. Event-based MILP models for resource-constrained project scheduling problems. *Computers & Operations Research* 38, 3–13. <http://dx.doi.org/10.1016/j.cor.2009.12.011>.
- Koné, O., Artigues, C., Lopez, P., Mongeau, M., 2013. Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. *Flexible Services and Manufacturing Journal* 25 (1-2), 25–47. <http://dx.doi.org/10.1007/s10696-012-9152-5>.
- Kopanos, G. M., Kyriakidis, T. S., Georgiadis, M. C., 2014. New continuous-time and discrete-time mathematical formulations for resource-constrained project scheduling problems. *Computers & Chemical Engineering* 68, 96 – 106. <http://dx.doi.org/10.1016/j.compchemeng.2014.05.009>.
- Levin, A., 1971. Scheduling and Fleet Routing Models for Transportation Systems. *Transportation Science* 5 (3), 232–255. <http://dx.doi.org/10.1287/trsc.5.3.232>.
- Li, Z., Guo, J., Zhou, R., 2016. Maintenance scheduling optimization based on reliability and prognostics information. In: *Annual Reliability and Maintainability Symposium (RAMS)*. pp. 1–5. <http://dx.doi.org/10.1109/RAMS.2016.7448069>.
- Liang, Z., Chaovalitwongse, W., 2013. A Network-Based Model for the Integrated Weekly Aircraft Maintenance Routing and Fleet Assignment Problem. *Transportation Science* 47, 493–507. <http://dx.doi.org/10.1287/trsc.1120.0434>.
- Martins, P. M. S., 2016. Systematization and optimization of fokker 100 s maintenance plan. Tech. rep., Instituto Superior Técnico, Lisboa, Portugal, accessed: 07/01/2020. https://fenix.tecnico.ulisboa.pt/downloadFile/3779578550788/PedroSena_ExtendedAbstract.pdf
- Mercier, A., Cordeau, J. F., Soumis, F., 2005. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers and Operations Research* 32 (6), 1451–1476. <http://dx.doi.org/10.1016/j.cor.2003.11.013>.
- Naber, A., 2017. Resource-Constrained Project Scheduling with Flexible Resource Profiles in Continuous Time. *Computers & Operations Research* 84, 33–45. <http://dx.doi.org/10.1016/j.cor.2017.02.018>.
- Nocedal, J., Wright, S., 2006. *Numerical Optimization: Springer Series in Operations Research and Financial Engineering*. Springer. <http://dx.doi.org/10.1007/978-0-387-40065-5>.
- Orhan, I., Kapanoglu, M., Karakoc, T. H., 2012. Concurrent Aircraft Routing and Maintenance Scheduling Using Goal Programming. *Journal of the Faculty of Engineering and Architecture of Gazi University* 27 (1), 11–26, accessed: 07/01/2020, https://www.researchgate.net/publication/256457265_Concurrent_Aircraft_Routing_and_Maintenance_Scheduling_Using_Goal_Programming.
- Papadakos, N., 2009. Integrated airline scheduling. *Computers and Operations Research* 36 (1), 176–195. <http://dx.doi.org/10.1016/j.cor.2007.08.002>.

- Qantas, 2016. Qantas News Room: The A, C and D of Aircraft Maintenance. Accessed: 07/01/2020. www.qantasnewsroom.com.au/roo-tales/the-a-c-and-d-of-aircraft-maintenance/
- Safaei, N., Jardine, A. K., 2018. Aircraft routing with generalized maintenance constraints. *Omega* 80, 111 – 122. <http://dx.doi.org/10.1016/j.omega.2017.08.013>.
- Sarac, A., Batta, R., Rump, C. M., 2006. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research* 175 (3), 1850–1869. <http://dx.doi.org/10.1016/j.ejor.2004.10.033>.
- Sousa, J. P., Wolsey, L. A., 1992. A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming* 54 (1-3), 353–367. <http://dx.doi.org/10.1007/BF01586059>.
- Sriram, C., Haghani, A., 2003. An optimization model for aircraft maintenance scheduling and re-assignment. *Transp. Res. Part A Policy Pract.* 37 (1), 29–48. [http://dx.doi.org/10.1016/S0965-8564\(02\)00004-6](http://dx.doi.org/10.1016/S0965-8564(02)00004-6).
- Yan, S., Tseng, C.-H., 2002. A passenger demand model for airline flight scheduling and fleet routing. *Computers and Operations Research* 29 (11), 1559–1581. [http://dx.doi.org/10.1016/S0305-0548\(01\)00046-6](http://dx.doi.org/10.1016/S0305-0548(01)00046-6).

Appendix A. Supplementary Proofs

Proposition Appendix A.1 For a given interval i and aircraft k , the following inequality holds,

$$\sup\{MOP_k^{j-1}\} < i \leq \sup\{MOP_k^j\},$$

for some MOP_k^j provided that $j \neq 1$, $i > \sup\{MOP_k^1\}$ and $i \leq \sup\{MOP_k^J\}$ ($J = |MOP_k|$). In other words, provided that the interval under consideration starts after the end of the first MOP and before the end of the last MOP.

Proof. For a given i and aircraft k , either, $i \in \mathcal{I}_k$ or $i \notin \mathcal{I}_k$.

In the first case, by definition $i \in MOP_k^j$ for some j , hence, clearly, $i \leq \sup\{MOP_k^j\}$. Also, by definition, intervals in MOPs are disjoint, $i \notin MOP_k^{j-1}$, thus $i > \sup\{MOP_k^{j-1}\}$. Therefore, the inequality holds.

Otherwise, we have,

$$i \notin \mathcal{I}_k \wedge i \leq \sup\{MOP_k^J\} \implies \exists j = \max\{j: j \in MOP_k, j \leq J, i < \inf\{MOP_k^j\}\}.$$

Hence, $i \leq \sup\{MOP_k^j\}$, and, therefore, $i > \sup\{MOP_k^{j-1}\}$. This completes the proof. \square

Proposition Appendix A.2 Transitivity constraints 11 – 13 in Model 3.2, ensure non-preemption for all MOPs.

Proof. We want to prove the following relationship,

$$\text{Transitivity constraints} \implies \text{Non-preemption},$$

or, equivalently,

$$\text{Preemption} \implies \text{Transitivity constraints do not hold.}$$

Suppose that, maintenance begins at an interval $i \in MOP_k^j$, or,

$$m_{kc}^{i-1} = 0 \wedge m_{kc}^i = 1 \text{ for some } i, i-1 \in MOP_k^j.$$

Using constraints 12, we have that $z_{kc}^i = 1$. Now, assume there is a preemption, that is,

$$m_{kc}^{i'-1} = 0 \wedge m_{kc}^{i'} = 1 \text{ for some } i' \in MOP_k^j, i' > i.$$

By constraints 12, we have that $z_{kc}^{i'} = 1$. However, given constraints 13 and $z_{kc}^i = 1$, we have a contradiction. Therefore, we must have

$$m_{kc}^{i'} - m_{kc}^{i'-1} \leq 0 \forall i' \in MOP_k^j, i' > i.$$

Hence, within an MOP, once maintenance has started, we can either continue, in which case $m_{kc}^{i'} - m_{kc}^{i'-1} = 0$, or end it, where $m_{kc}^{i'} - m_{kc}^{i'-1} = -1$. We cannot start another maintenance intervention within the same MOP.

For the boundary condition, if i is the first interval in the MOP, $i = \inf\{MOP_k^j\}$, then by constraints 11 and 13, the claim still holds. This completes the proof. \square