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 re-assignments, number of maintenance interventions, overall resource usage, and number of not airworthy aircraft. 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 (6457 flights, 1032 aircraft, 5 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 economical 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:

1. Airframe checks (A, B, C and D);

*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)

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, 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).

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
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 while adhering to the airlines' flight schedule. 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.

Although optimisation models that study aircraft maintenance have been developed, certain aspects are not been fully addressed. Such models, mainly produce cyclic maintenance schedules for short-term maintenance; and, albeit considering a single airline, no individual workshop restrictions, or legal remaining flying hours; they tend to be inefficient even for small–medium data sets. In this paper we introduce a framework that addresses these gaps. The proposed framework deals with the requirements introduced by a 30-day planning horizon, multiple airlines, individual legal remaining flying hours, and tight resource availabilities. Producing adaptive maintenance schedules for large data sets while remaining computationally efficient.

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 the

AMS problem with TA. Section 4 presents the solution methodology and the model application and computational tests, while Section 5 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); tail assignment (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) (Gopalan and Talluri, 1998).

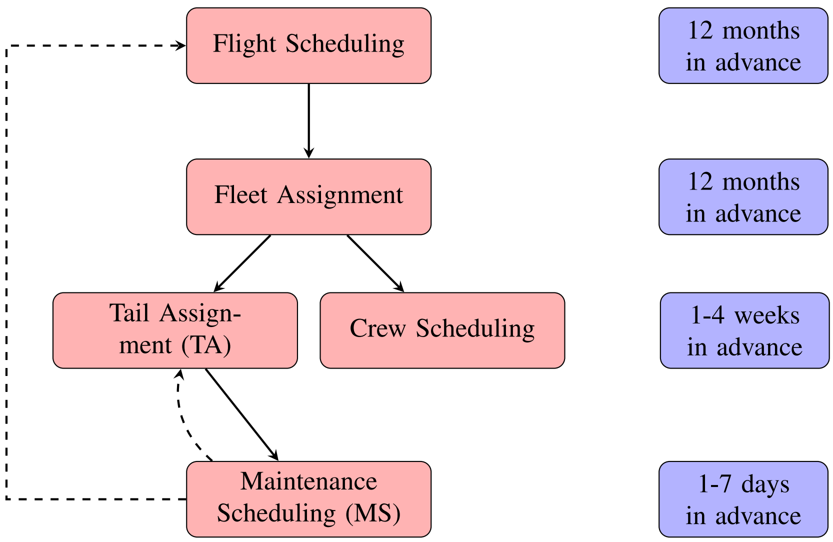


Figure 1: Stages of the airline scheduling problem.

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

String-based models: a type of formulation that formulates 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: as can be seen in Figure 2, in a TSN network, each airport (A and B) 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);

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. The formulation, typically, includes capacity constraints (passengers and fleet) and conservation of aircraft, flight, and airport flow (Levin, 1971).

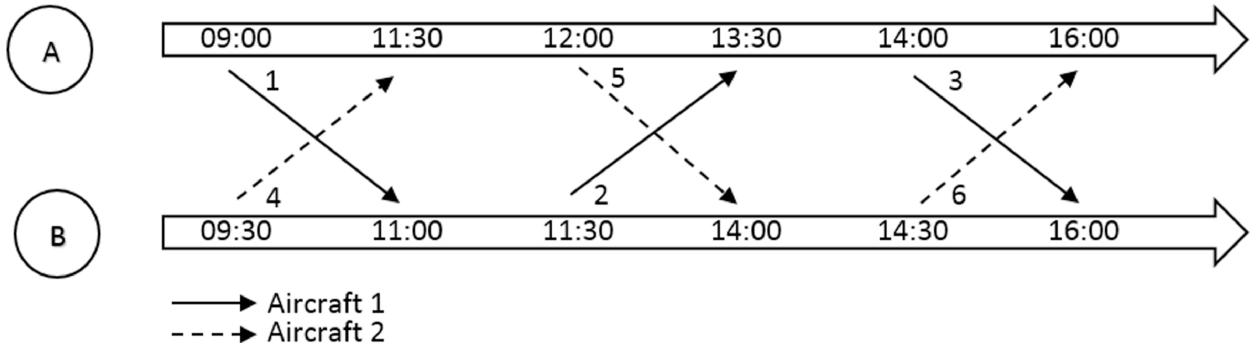


Figure 2: Time-space network from Khaled et al. (2018).

Desaulniers et al. (1997) and Barnhart et al. (1998), published their pioneering work on formulating the FA and the TA using a string-based formulation and solving it using branch and bound. Even though they include maintenance regulations, they did not consider workshop resources or flying hours (FH). Sarac et al. (2006); Cohn and Barnhart (2003); Papadakos (2009) use the same type of formulation to solve the AMR. The work reported in 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 Chaovalitwongse, 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) examine the AMR problem and include 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 Feo and Bard (1989) formulation of the TA, which solves the AMR incorporating A and B checks. They used a heuristic algorithm to solve a small test instance (58 flights, 75 airports) in 5 minutes.

All the publications mentioned are limited to daily or weekly schedules and assume cyclical repetitions of the flights and hence maintenance operations. Khaled et al. (2018), however, caters to individual maintenance requirements for a 30-day plan. Using an improved TSN formulation for the AMR problem, they effectively schedule A checks, constrained by individual aircraft legal remaining FH. Furthermore, they assume that maintenance is generally performed at night. Due to the type of formulation, solution times increase noticeably for a large number of flights (timed out at 3 hours for 1494 flights and a single airline). Li et al. (2016), deal with AMR for fighter jets. This formulation, relies on a single workshop assumption, also employs legal remaining FH to determine the frequency of maintenance. Further assumptions include, 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 25 days and 200 aircraft but no solution times are provided. However, the timeline is discretised by splitting each day into two 12 hours 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. The objective is, usually, to minimise the duration of the project (collection of ordered jobs). The reader may refer to a recent and thorough review article, Habibi et al. (2018). On the long-term RCPSP, Koné et al. (2011) propose two formulations which they refer to as "event-based RCPSP". Events corresponding 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, 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	H	Sched. Type	Airlines	F	A	T
Desaulniers et al. (1997)	×	×	×	1	cyclical	single	383	91	1 h
Barnhart et al. (1998)	✓	×	×	7	cyclical	single	1124	89	10 h
Yan and Tseng (2002)	×	×	×	1	cyclical	single	170	17	2546 s
Sriram and Haghani (2003)	✓	×	×	7	cyclical	single	rand.	58	5 mins
Mercier et al. (2005)	✓	×	×	1	cyclical	two	707	143	13 h
Sarac et al. (2006)	✓	✓	✓	1	cyclical	single	175	32	2 h
Papadakos (2009)	✓	×	×	7	cyclical	single	705	167	16 h
Haouari et al. (2013)	✓	×	×	1	cyclical	single	344	138	8.46 s
Liang and Chaovalitwongse (2013)	✓	×	×	7	cyclical	single	1780	110	4 h
Khaled et al. (2018)	✓	✓	×	30	adaptive	single	1494	40	3 h
Present paper	✓	✓	✓	30	adaptive	multiple	6457	1032	1.5 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 and resource considerations, while providing fast solution times. Additionally, we take into account individual aircraft airworthiness and their respective flight operations. The key contributions of the present paper are,

Tail re-assignment. We include the option of re-assigning some flights to obtain longer maintenance opportunities (MOPs).

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 flying hours. 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.

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. To check if regulation requirements are being fulfilled, we employ different airworthiness (AW) variables for each type of maintenance and aircraft type, which vary with flying hours (FH).

In the first formulation, given an input flight schedule, we aim to determine whether there exists a feasible maintenance schedule. We call this the the airline fleet maintenance scheduling (AMS) formulation. If a feasible maintenance schedule does not exist, we want 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 tail assignment (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 maintenance opportunities (MOPs);
2. Maintenance can be done at any workshop, if the aircraft has a MOP there;
3. To extend MOPs, we may assign flights to different aircraft, i.e. we may modify the airlines' preferred tail assignment;
4. Resources can be shared amongst different airlines at the maintenance workshops.

3.2. *Concepts and Notation*

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 both 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 (RCPSP) 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 RCPSP". Events corresponding to start and/or end times of activities. Compared to traditional time indexation, their formulations involve considerably fewer variables, therefore, being ideal for problems with long planning horizons. Moreover, Koné et al. (2013) extend the formulation to account for non-renewable resources. More authors have used a similar indexation of continuous time. Naber (2017) focuses on removing the assumption regarding fixed resources per activity in the RCPSP by allowing flexibility of resource usage.

In this paper, events represent turnaround times, time between arrival and departure, at a certain maintenance workshop where the total time is sufficient to perform at least one maintenance type. We regard these as MOPs. Provided with a flight schedule, MOPs are easily identified and pose no restrictive assumption on the aircraft or airlines considered.

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 $t : \mathcal{K} \to \mathcal{T}$; $t(k)$ 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. Intervals provide an alternative discretisation of time and represent either a start or end time of some MOP. All time intervals are contained in the set $\mathcal{I}$. Each element, i, has a start and end time, denoted with st^i and et^i respectively, and a maintenance workshop W^i. All flights are contained in the set $\mathcal{F}$. More precisely, it contains flight legs (sequence of multiple flights) between MOPs. Some common maintenance types are collected in the set C. Resources are contained in the set R. The demand for resource $r \in R$ varies per check $c \in C$, 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 $c \in C$ to take $\Delta_{t(k),c}$ time units to complete.

For convenience, we can identify some subsets. Let $\mathcal{I}_k$ be the subset of intervals for aircraft k, and $\mathcal{I}_f$ be the subset of intervals at which flight $f \in \mathcal{F}$ occurs. Also, $\mathcal{K}_i$ and $\mathcal{K}_f$ represent the set of aircraft available at interval i and flight f respectively. 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 $i \in MOP_k^j$. The simple idea is illustrated in Figure 3, where an example for an aircraft k

with three MOPs is shown. Each of MOP is labelled with a different $j = 1, 2, 3$. Lastly, for a given interval i, we can identify the set of flights that depart at interval i. Let us denote this set by

$$\mathcal{F}_{=i} = \left\{ f : f \in \mathcal{F},\ \inf\left(\mathcal{I}_f\right) = i \right\} .$$

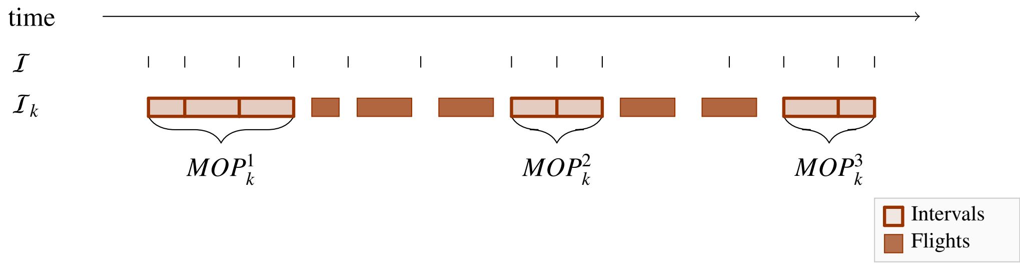


Figure 3: Timeline showing the deconstruction of three MOPs into sets of consecutive intervals.

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, briefly discussed in the literature review section, we define a variable, $w_{kc}^i \in [0, 1]$, that tracks AW for aircraft $k \in \mathcal{K}_i$ at the beginning of interval i, and check type c. We define two binary variables to identify regulation violations in the two models. In the first model, 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. Similarly, in the second model, v_k^f gets the value 1 if a regulation is being violated before flight f for aircraft k, while it is 0 otherwise.

3.3. *Definitions*

Sets

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

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

$\mathcal{F}_{=i}$**:** Set of flights which are scheduled in interval i;

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

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

$\mathcal{I}_f$**:** Set of intervals occupied by flight f, $\mathcal{I}_f \subseteq \mathcal{I}$;

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

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

$\mathcal{K}_f$**:** Set of aircraft available to fly flight f, $\mathcal{K}_f \subseteq \mathcal{K}$;

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, $MOP_k^j \subseteq \mathcal{I}_k$;

R**:** Set of resources indexed by r;

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

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

Parameters

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

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

DR^f_{kc}: Deterioration rate for check c for aircraft k for the operation of flight f;

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

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

Variables

a^f_k: 1, if the scheduled flight f is assigned to aircraft k; 0, otherwise.

B^i_r: The capacity for resource r at interval i.

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

w^i_{kc}: A continuous variable with values between 0 and 1 to represent the AW for aircraft k for check c at the beginning of interval i. 0 means that the aircraft requires no maintenance and is airworthy, 1 means that the aircraft requires maintenance urgently and is not airworthy.

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

v^f_k: 1, if a regulation is being violated before flight f for aircraft k; 0, otherwise.

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

z^i_{kc}: 1, if there is a change between the consecutive variables, m^i_{kc} and m^{i-1}_{kc}; 0, otherwise.

3.4. Airline Fleet Maintenance Scheduling with Violations

Model 3.1. Interval MMILP formulation for AMS with violations.

$$\textbf{min} \quad \sum_{i} \sum_{c} \sum_{k \in \mathcal{K}_i} v_{kc}^{i} \tag{1}$$

$$\textbf{min} \quad \sum_{c} \sum_{k} w_{kc}^{\sup\{\mathcal{I}_k\}} \tag{2}$$

Subject to

Airworthiness

$$w_{kc}^{i} \geq w_{kc}^{i-1} + DR_{kc}^{i}\left(1 - m_{kc}^{i}\right) - m_{kc}^{i} - v_{kc}^{i} \qquad \forall i \in \mathcal{I} \setminus \inf\{\mathcal{I}\},\ c,\ k \in \mathcal{K}_i; \tag{3}$$

$$w_{kc}^{i} \leq DR_{kc}^{i+1} \qquad \forall i \in \mathcal{I} \setminus \sup\{\mathcal{I}\},\ c,\ k \in \mathcal{K}_i; \tag{4}$$

Maintenance

$$m_{kc}^{i} = 0 \qquad \forall i,\ c,\ k \notin \mathcal{K}_i; \tag{5}$$

$$\sum_{c} m_{kc}^{i} \leq 1 \qquad \forall i,\ k \in \mathcal{K}_i; \tag{6}$$

$$\sum_{i' \in MOP_k^j} (et^{i'} - st^{i'}) m_{kc}^{i'} \geq \Delta_{t(k),c} m_{kc}^{i} \qquad \forall k,\ c,\ j,\ i \in MOP_k^j; \tag{7}$$

Transitivity Constraints

$$z_{kc}^{\inf\left\{MOP_k^j\right\}} \geq m_{kc}^{\inf\left\{MOP_k^j\right\}} \qquad \forall k,\ c,\ j; \tag{8}$$

$$z_{kc}^{i} \geq m_{kc}^{i} - m_{kc}^{i-1} \qquad \forall k,\ c,\ j;\ i, i-1 \in MOP_k^j; \tag{9}$$

$$\sum_{i \in MOP_k^j} z_{kc}^{i} \leq 1 \qquad \forall k,\ c,\ j; \tag{10}$$

Resources

$$\sum_{k \in \mathcal{K}_i} \sum_{c} b_{rc} m_{kc}^{i} \leq B_r^{i} \qquad \forall i,\ r \in R_{W^i}; \tag{11}$$

$$B_r^{i} \leq \overline{B}_r \qquad \forall i,\ r \in R_{W^i}; \tag{12}$$

Variables

$$w_{kc}^{i} \in [0, 1] \qquad \forall i,\ c,\ k \in \mathcal{K}_i; \tag{13}$$

$$B_r^{i} \in \mathbb{R}^{+} \qquad \forall i,\ r \in R_{W^i}\ k \in \mathcal{K}_i; \tag{14}$$

$$\overline{B}_r \in \mathbb{R}^{+} \tag{15}$$

$$m_{kc}^{i},\ z_{kc}^{i},\ v_{kc}^{i} \in \{0, 1\} \qquad \forall i,\ c,\ k \in \mathcal{K}_i. \tag{16}$$

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 number of aircraft that are not airworthy at the last interval. Objective 1 minimises the number of regulation violations. Objective 2 minimises the number of aircraft that are not airworthy at the end of the planning horizon. Recall that a value close to 0 for the w_{kc}^{i} variable indicates that the aircraft is airworthy; hence, minimising $w_{kc}^{\sup\{\mathcal{I}_k\}}$ corresponds to minimising the number of aircraft that are not airworthy at the end of the planning horizon.

The AW constraints, enforce an iterative relation captured in Constraints 3. For the time interval 1, for any aircraft k and check type c, w_{kc}^{1} is updated using the initialisation w_{kc}^{0} plus the deterioration rate DR_{kc}^{1} if maintenance is not performed, or drops to 0 if either a maintenance or violation occurs. Similarly, w_{kc}^{i} is updated using w_{kc}^{i-1}, for all i. Also, Constraints 4 ensure that the AW variable stays within the regulation limits (captured within the deterioration rate parameter) at least until the next MOP.

Constraints 5 – 7 are the maintenance constraints. Constraints 5 ensure that maintenance is not performed when an aircraft is operating flights. Constraints 6 assure that no more than one maintenance type is scheduled for the same interval. Constraints 7 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 a check.

Transitivity constraints, 8, 9, and 10, ensure that if we decide to maintain in MOP_k^j, preemptions are not allowed. Constraints 8 initialise the auxiliary variable using the first interval in the MOP. Constraints 9 establish that when a maintenance starts, i.e. the difference between consecutive maintenance variables is 1, the auxiliary variable is 1. Constraints 10 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 11 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. Constraints 12 establish the value for the maximum resource level for each resource. The last four constraints 13 – 16 define the variables.

3.5. Airline Fleet Maintenance Scheduling with Tail Assignment

In this section we extend the AMS formulation previously discussed, to include TA 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 generate TA variables and solve the joint AMS and TA problem formulated below. More details on how TA variables are generated is given in Section 4.2.

Model 3.2. Interval MMILP formulation for AMS with TA.

$$\textbf{min} \quad \sum_{f} \sum_{k \in \mathcal{K}_f} v_k^f \tag{17}$$

$$\textbf{min} \quad \sum_{r} \overline{B}_r \tag{18}$$

$$\textbf{min} \quad \sum_{f} \sum_{k \in \mathcal{K}_f} a_k^f \tag{19}$$

$$\textbf{min} \quad \sum_{i} \sum_{c} \sum_{k \in \mathcal{K}_i} \left(et^i - st^i\right) m_{kc}^i \tag{20}$$

$$\textbf{min} \quad \sum_{i} \sum_{r} \left(et^i - st^i\right) B_r^i \tag{21}$$

$$\textbf{min} \quad \sum_{c} \sum_{k} w_{kc}^{\sup\{\mathcal{I}_k\}} \tag{22}$$

Subject to

Airworthiness

$$w_{kc}^i \geq w_{kc}^{i-1} + \sum_{f \in \mathcal{F}_{=i}} \left(DR_{kc}^f a_k^f - v_k^f\right) - m_{kc}^i \qquad \forall i \in \mathcal{I} \setminus \inf\{\mathcal{I}\},\ c,\ k \in \mathcal{K}_i; \tag{23}$$

$$w_{kc}^i \leq DR_{kc}^f \qquad \forall i,\ c,\ k \in \mathcal{K}_i,\ f \in \mathcal{F}_{=i}; \tag{24}$$

Tail Assignment

$$m_{kc}^i \leq 1 - \sum_{f \in \mathcal{F}_{=i}} a_k^f \qquad \forall i,\ c,\ k; \tag{25}$$

$$\sum_{k} a_k^f = 1 \qquad \forall f; \tag{26}$$

Maintenance

5 – 7

Transitivity Constraints

8 – 10

Resources

11 and 12

Variables

$$a_k^f, v_k^f \in \{0, 1\} \qquad \forall k,\ f; \tag{27}$$

13 – 16

The proposed formulation is a MMILP with six lexicographically ordered objective functions. The functions involved minimise the following objectives (in order of importance), the number of violations, maximum resource level, number of tail re-assignments, number of maintenance interventions, overall resource usage, and number of not airworthy aircraft. Objective 17, with largest priority, minimises the number of regulation violations. Objective 18 minimises the sum of maximum level for each resource. Objective 19 minimises the number of tail reassignments. Objective 20 minimises the number of maintenance interventions weighted using the duration of the interval. Objective 21 minimises the resource level per interval, again, weighted with the duration of intervals. Finally, as in the previous model, objective 22, with least priority, minimises the the number of aircraft that are not airworthy by the last interval.

The AW constraints are a simple extension of those in Model 3.1. Constraints 23 enforce a recurrence relation where w_{kc}^i is updated using w_{kc}^{i-1} plus a deterioration term only if the aircraft has been assigned the flights during the interval under consideration. In the case where aircraft k is assigned to flight f which is at i, $a_k^f = 1$, and thus the AW increases at the appropriate rate for the flight, DR_{kc}^{f}. Moreover, constraints 24 ensure that no regulations are being violated for any flight after interval i, at least until the next MOP.

As for the tail assignment, constraints 25 limit maintenance to the intervals where aircraft has not been assigned a flight. Constraints 26 ensure that all flights have one aircraft assigned to them.

The remaining constraints, (maintenance, transitivity and resource constraints) as well as variable definitions, can be borrowed from Model 3.1. The last constraints 27, define the extra variable.

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 solution process, after a preprocessing routine, we are implementing 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 an increasingly larger size for the tail assignment (TA) problem 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. This is a more efficient approach than starting with too many short time intervals. The proposed approach increases the computational efficiency without sacrificing accuracy.

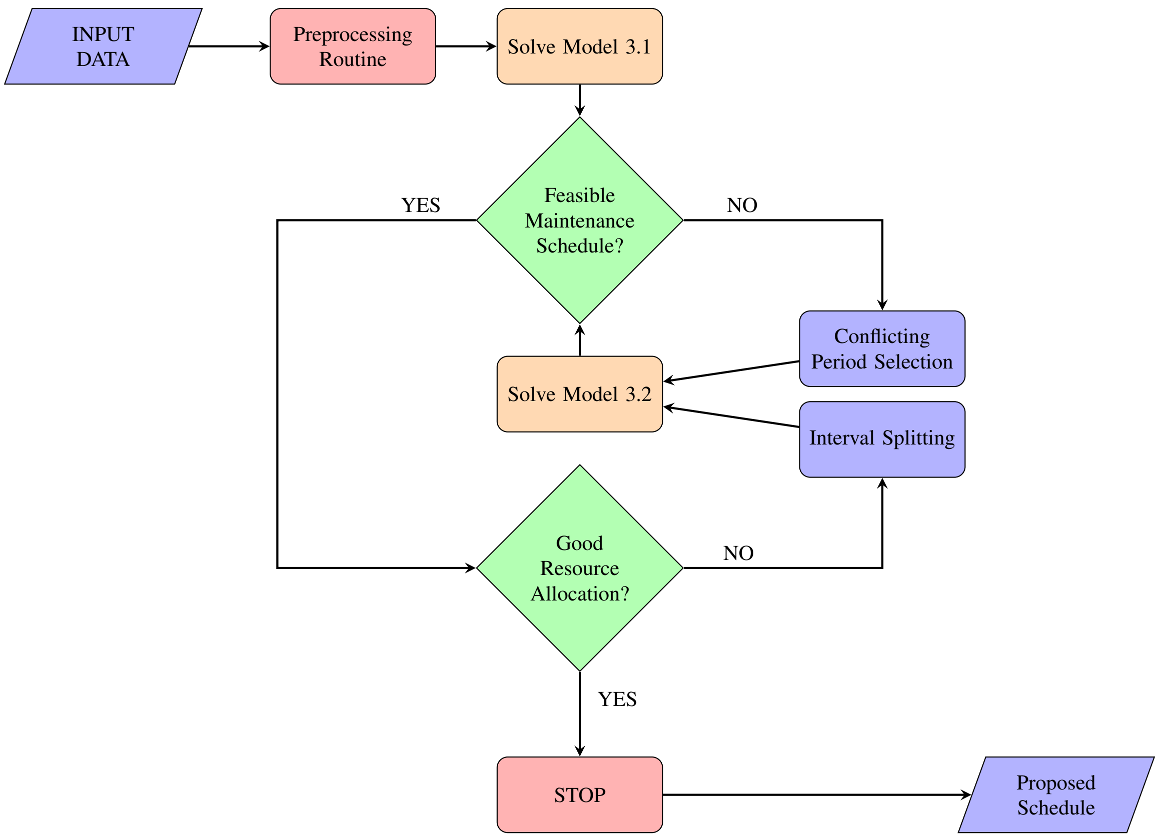


Figure 4: Flow chart outlining the process of the iterative algorithm.

4.1. *Preprocessing Routine*

Flight schedules are crucial for the model as they give the initial TA and accurate flying hours (FH) in order to update the airworthiness throughout the planning horizon. We obtained flight schedules from Flightradar24 AB (2018) using the package `pyflightdata` in Python (Allamraju, 2014). We gathered data globally for an extended period of time. 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 five maintenance workshops over a 30-day planning horizon (between dates 14/11/16 and 15/12/16), we have 6457 flights and 1032 aircraft of two types (Airbus A320 and Fokker 100). Then, we proceed to identify maintenance opportunities (MOPs) i.e. turnaround times sufficient to perform at least the shortest maintenance type. We choose a turnaround time of 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.

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 size of the TA. In each iteration, if the regulation violations involved have not been removed, we increase the size of the conflicting period (which determines the number of TA variables) and resolve Model 3.2. In the interval splitting stage, once all infeasibilites have been removed or the largest TA has been solved, 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 $\{k_1, k_2, k_3, \dots, k_m\} \in \mathcal{K}_i$, we select the initial conflicting period as shown in red with $j^i_{(1)}$ as the upper bound, provided that $\mathcal{K}_i \subseteq \mathcal{K}_{j^i_{(1)}}$. We generate TA variables for the flights and aircraft involved in the conflict and solve Model 3.2. If in the new solution, interval i still has as a violation, we increase the size of the conflicting period to reach $j^i_{(2)}$ (provided that $\mathcal{K}_i \subseteq \mathcal{K}_{j^i_{(2)}}$). Again, we generate the appropriate TA variables 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 the largest TA would have been solved.

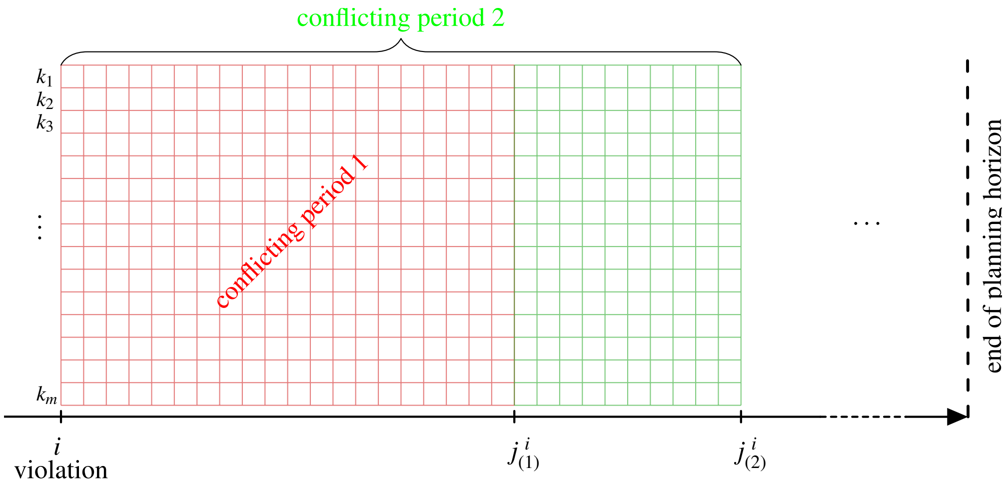


Figure 5: Conflicting period selection.

In order to identify the conflicting period, we have to identify where violations occur. For this purpose, consider the set of flights,

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

Now, let set V contain all intervals for which at least one violation occurs. That is,

$$V = \left\{ i : i \in \mathcal{I}_f, f \in F \right\} .$$

For every 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 $\mathcal{K}_j$, $\mathcal{K}_i \subseteq \mathcal{K}_j$, where $j > i$. Hence, we only need to generate TA variables for the flights between intervals i and j, the conflicting period, and for $k \in \mathcal{K}_i$. The set of intervals J_i which contain the set of aircraft $\mathcal{K}_i$ can be written as,

$$J_i = \{ j : j \in \mathcal{I}, i < j, \mathcal{K}_i \subseteq \mathcal{K}_j \} . \tag{28}$$

With this, we can write the first conflicting period as

$$CP(1, i) = \{ i' : i' \in \mathcal{I}, i \leq i' \leq j^{\,i}_{(1)}, \exists k \in \mathcal{K}_{i'} \wedge k \in \mathcal{K}_i \} , \tag{29}$$

where $j^{\,i}_{(1)}$ is the first element in J_i. We use this set to generate the appropriate variables for the TA problem. To do so, we only 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 this update, we can assign 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 $m = 2, \ldots, |J_i|$, which uses uses the m-th element of J_i, $j^{\,i}_{(m)}$. Similarly, to generate the TA variables, we set

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

We do this for every violation $i \in V$, until either the violation is eliminated or the largest TA has been solved.

4.2.2. Interval Splitting

The interval splitting stage favours the redistribution of resources by using a more granular timeline in each iteration. Resources are occupied for the duration of the interval if maintenance is being performed. In the case where intervals are long, it may be that resources are being 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.

Let us define three simple splitting methods. Suppose, after solving the problem, we have $m^i_{kc} = 1$, 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.

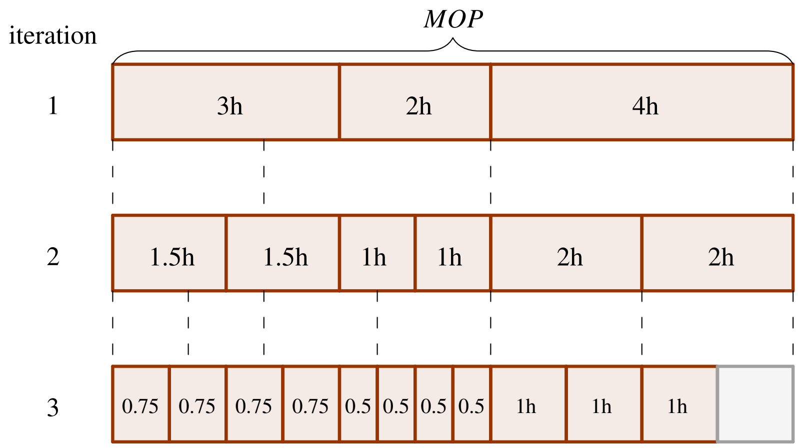


Figure 6: Interval splitting stage for a 9 hour MOP using binary segmentation for three iterations.

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,

$$A = \frac{\sum_c \sum_k \sum_j \Delta_{t(k),c}}{\sum_c \sum_k \sum_j \sum_{i \in MOP^j_k} (et^i - st^i) m^i_{kc}} .$$

Given constraints 7, 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; given a set of intervals, set of aircraft, FH for each aircraft, maintenance regulations and durations, and resource capacities and demands; we proceed in the following fashion. 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.

```
Initialisations
iter, A = 0                                   ▷ Iteration counter and accuracy measure
ε = 0.1                                       ▷ Set Tolerance
maxiter = 50                                  ▷ Maximum number of iterations
Intervals = I                                 ▷ Interval set
V, CP, oldIntervals = [ ]                     ▷ Empty array for violations, conflicting periods, and old intervals
while iter < maxiter and A < 1 − ε do
    if iter = 0 then
        Solve Model 3.1                       ▷ For the first iteration
    else
        Solve Model 3.2                       ▷ For iterations ≥ 1
    end if
    Update V and A                            ▷ Update violations and accuracy measure using new solution
    if V is not empty then                    ▷ If violations have not been removed
        SelectCP(V)                           ▷ Call function to update conflicting periods
    else
        which = {i : i ∈ I, Σ_k Σ_c m^i_{kc} ≥ 1}   ▷ Identify intervals with maintenance scheduled
        oldIntervals= Intervals
        SplitIntervals(which, Intervals)      ▷ Call function to split intervals
        if |Intervals| = |oldIntervals| then  ▷ No intervals have been added
            Break                             ▷ Stop the algorithm
        end if
    end if
    Increment iter
end while

function SelectCP(V)                          ▷ Updates the conflicting periods for all violations
    for i in V do
        Compute J_i
        if i not in CP then                   ▷ If the violation is new
            CP = CP ∪ CP(1,i)                 ▷ Update CP with first conflicting period
        else                                  ▷ If violation has occurred at a previous iteration
            for m in 2 → |J_i| do             ▷ Iterate through elements in J_i
                if CP(m-1,i) in CP then       ▷ Find already used CP
                    CP = CP \ CP(m-1,i)       ▷ Remove it from conflicting periods
                    CP = CP ∪ CP(m,i)         ▷ Update CP with the next, m-th conflicting period
                    Break
                end if
            end for
        end if
    end for
    Intervals = Intervals ∪ CP                ▷ Update intervals
end function

function SplitIntervals(which, Intervals)     ▷ Splits intervals in which
    for i in which do
        Using Method 1, 2, or 3; split i into i_1 and i_2
        Intervals= Intervals\{i} ∪ {i_1, i_2}  ▷ Update Intervals
    end for
end function
```

4.3. 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 are for five maintenance workshops between dates 14/11/16 and 15/12/16. The iterative algorithm was written in `Python`, using Gurobi Optimization version 8.0 (2018) to solve the models, and implemented in parallel for each maintenance workshop. 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 269 flights and 239 aircraft, produces 389 intervals;

Bangkok Suvarnabhumi Airport: with 5720 flights and 357 aircraft, produces 843 intervals;

Cairo International Airport: with 195 flights and 182 aircraft, produces 279 intervals;

Dubai International Airport: with 181 flights and 213 aircraft, produces 63 intervals;

Tokyo Haneda International Airport: with 92 flights and 41 aircraft, produces 162 intervals.

As for the experimental set-up, we restrict the interval splitting stage of the algorithm such that the resulting intervals are at least 10 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. Resource demand and capacities, shown in Table 3, 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 initial AW variable is sampled from a Uniform distribution as follows,

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

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. Moreover, the formulations are solved using lexicographic ordered objectives with priorities as suggested by practitioners.

Table 3: 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}	B_r^i
r_1	1	1	25
r_2	3	5	25
r_3	2	3	25
r_4	1	2	25

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. The results for the workshops under consideration are shown in Table 4, while Figure 7 breaks down the objective function for the largest workshop. Specifically, Table 4 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 4, 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 88 minutes to terminate the algorithm. For the remaining, small to medium sized workshops, it takes between 0.1 seconds to 6 minutes to reach a good solution. Solution times for Method 1 and Method 2, are significantly larger with the former showing lower solution times.

The accuracy measure per iteration appears as a plot in the second column in Table 4. As can be seen, the accuracy measure evolves differently depending on the workshop. It takes a varying number of iterations across workshops to reach the required level of accuracy. In some cases, like Atlanta, such level is not reached; the algorithm terminates due to no more intervals being created. For the cases of Dubai and Tokyo, using Method 3, reaches the required accuracy level in very few iterations; whereas the rest take slightly more. Typically, neither Methods 1 or 2 reach the required level, with, Method 1 consistently providing better solutions.

A breakdown of the different components of the objective function value for the Bangkok workshop is given in Figure 7. The plot shows three stacked bars at each iteration, one for each splitting method, with the values of the objective function. At each iteration, the first bar is for Method 1, second bar for Method 2, and third bar for Method 3. The red component, `Bsum`, corresponds to the resource level; and, the blue component, `msum`, corresponds to the maintenance. Please note that only two objectives are shown, the reason for this is that, in comparison, the other objectives have negligible values; and, hence, have been omitted. The objectives with largest values are the fourth and fifth priority objectives, the weighted sum of the resource level, objective 20, and maintenance, objective 21. It can be seen that the objective values decrease for all the splitting methods with Method 3 providing the lowest objective value by iteration 4. This reinforces our assurance in the accuracy measure as it reveals that a lower objective function corresponds to a higher accuracy measure.

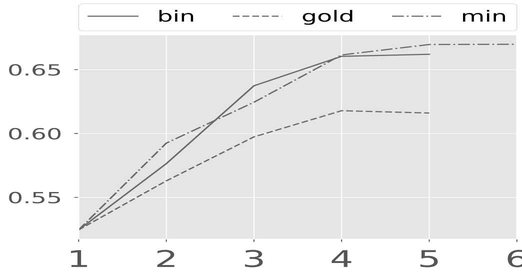
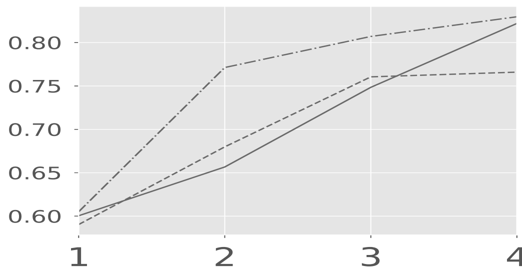
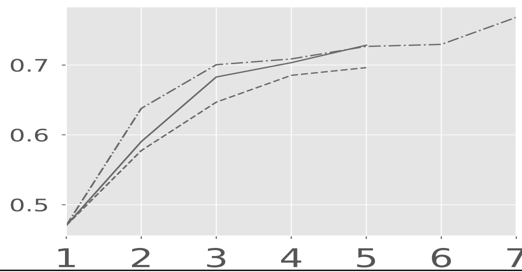
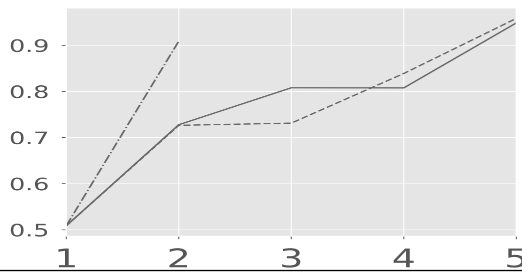
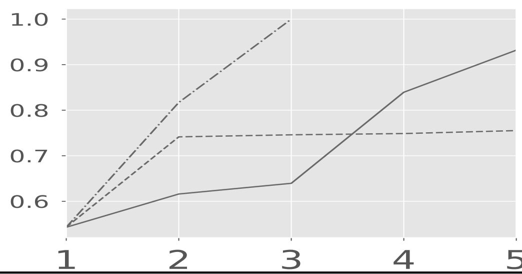
Due to its good solution times, lowest objective value, and overall higher accuracy measure we can claim that the interval splitting stage performs better using Method 3. Furthermore, it is worth noting that the aircraft are, also, kept in a better state at the end of the planning horizon. For this reason,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

Figure 8 shows two resource profiles (first and last iteration) for resource r_1. We can see that the resource profiles become considerably less populated and thinner. This means that the resources usage is smaller. Particularly, by the last iteration, the solution tend to have slightly higher resource levels so as to avoid performing maintenance during busy flight periods; hence, reducing the tail assignment objective. Moreover, as also seen in Figure 9, there is a build up of maintenance at the beginning of the planning horizon due to initialisation of the AW variable. However, this is quickly smoothed out and we suspect this would not occur if the variable was initialised using real AW values.

Figure 9 shows the maintenance schedule produced by the last iteration. The x-axis shows the date and the y-axis shows different aircraft tail numbers, which are plotted using different colours. In this plot we can see when the maintenance is being scheduled. It appears that most maintenance occurs during night and early morning shifts, as one would expect; with the exception of the 1st of December (1/12) when the only check of type 2 is also scheduled.

Since terminating the algorithm when $A \geq 1 - \varepsilon$, does not guarantee an optimal solution we conducted some further testing. For the Bangkok workshop, we split intervals every 5 minutes and solved the problem once (without time restriction) and compared the the solution with the one which we obtained using Method 3 by iteration 4. This revealed the same objective value in a considerable longer time; over 7 hours.

Table 4: 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		Final # intervals	2923	2817	925
		Total CPU time (min)	83	89	6
Bangkok		Final # intervals	5659	5401	559
		Total CPU time (min)	99	102	88
Cairo		Final # intervals	2383	2207	361
		Total CPU time (min)	59	64	5
Dubai		Final # intervals	473	425	85
		Total CPU time (s)	2.194	2.194	0.1
Tokyo		Final # intervals	531	537	191
		Total CPU time (s)	9.65	53	3.3

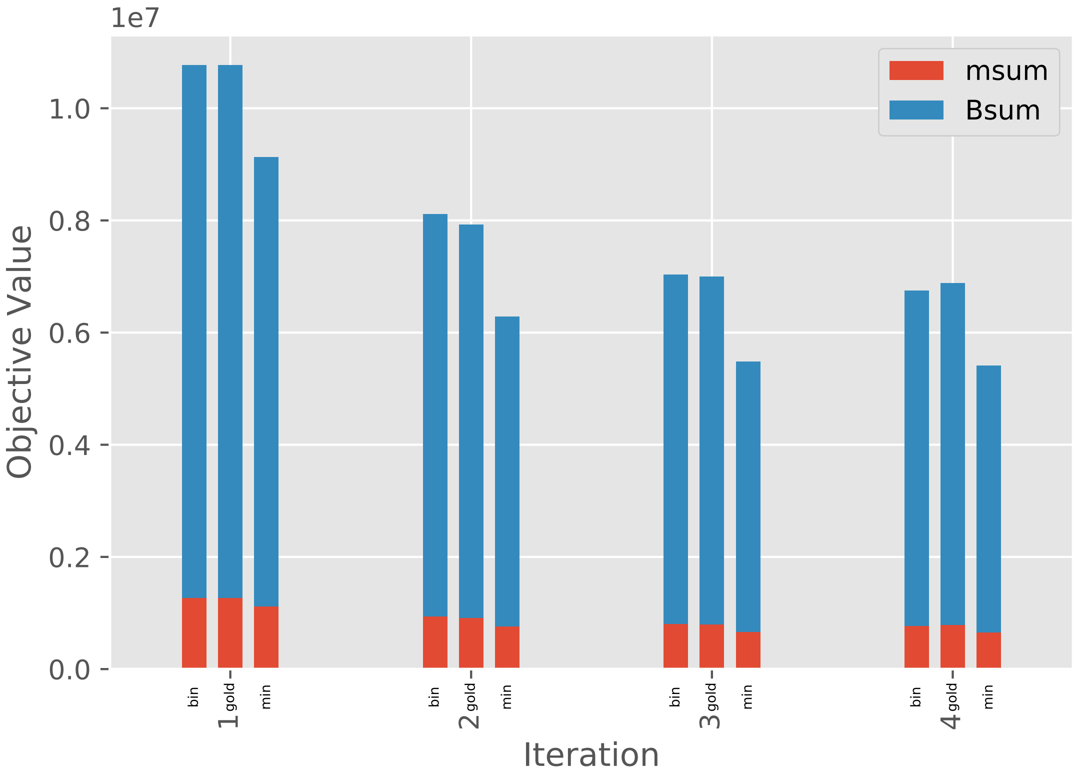
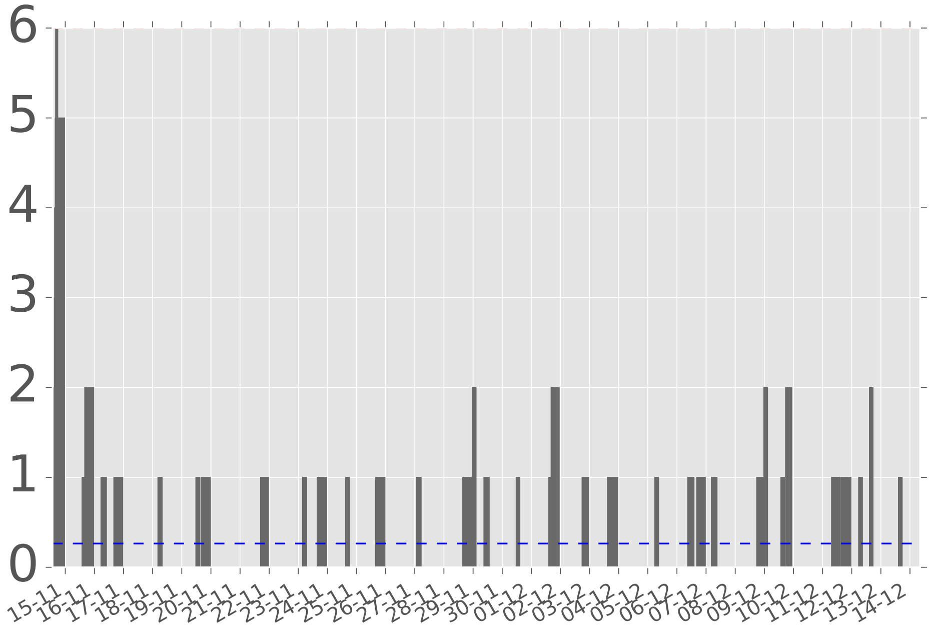
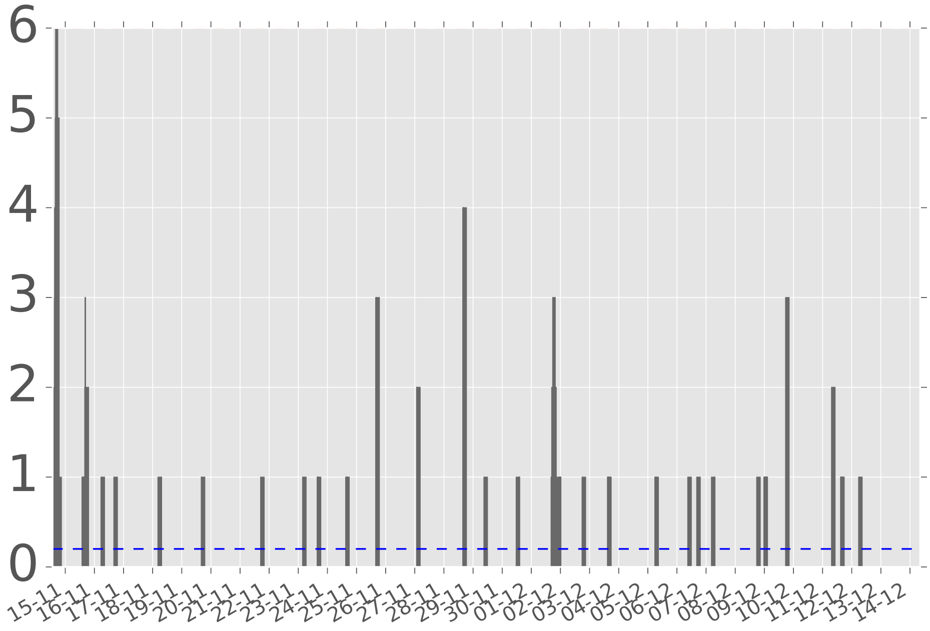


Figure 7: Components of the objectives against iterations for the three splitting methods for the workshop in Bangkok.



(a) Resource profile for the first iteration.



(b) Resource profile for the last iteration.

Figure 8: Resource profiles for the Bangkok workshop for the first and last (4th) iterations of the interval splitting stage using Method 3.

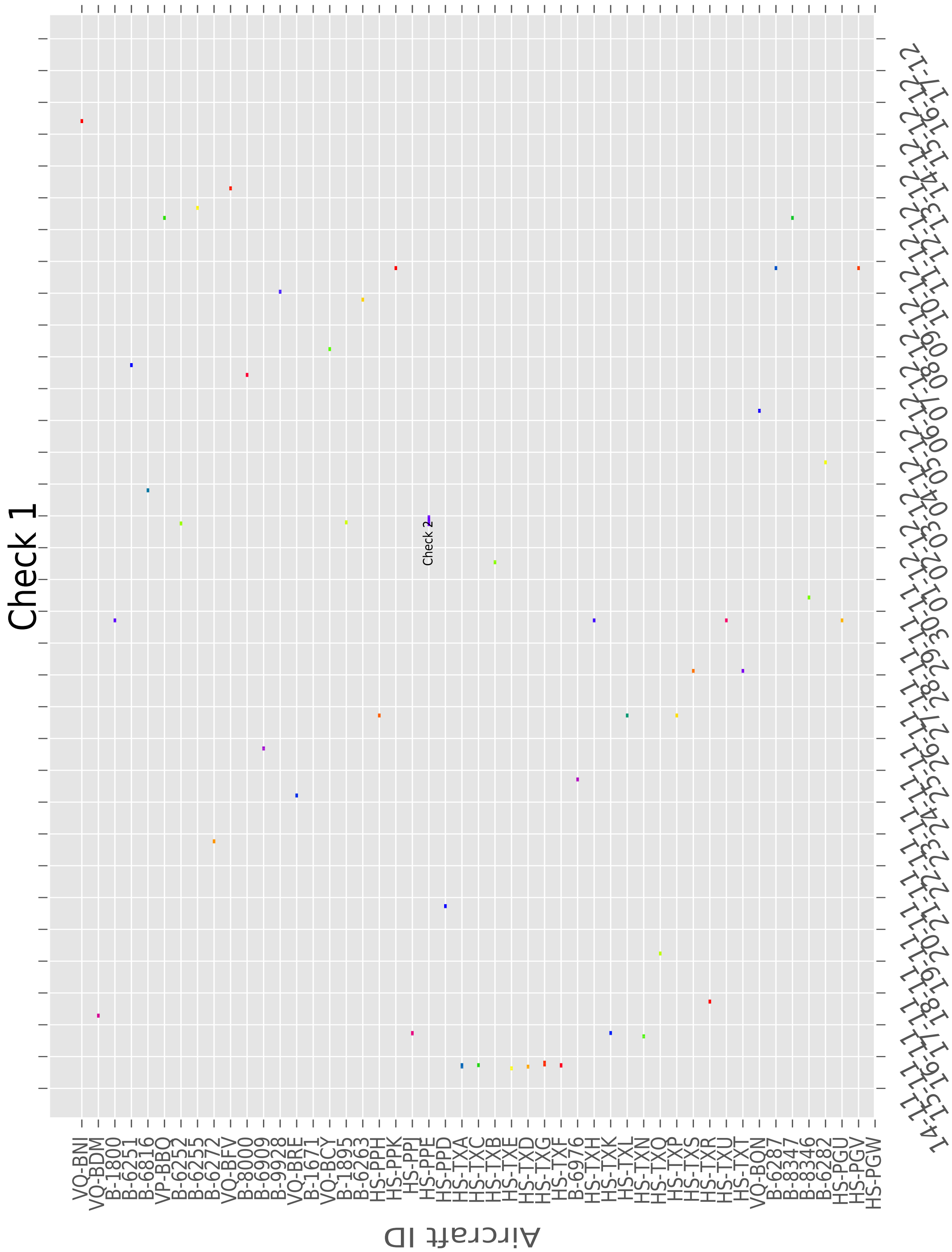


Figure 9: A selection of the maintenance schedule for the Bangkok case study. produced by the last iteration (4th) using Method 3.

5. 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 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. 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 airworthy, while remaining commercially viable for the airlines.

Some limitations are worth noting. Stopping the iterative algorithm when the accuracy measure is maximal does not guarantee an optimal solution. Clearly, if continued the algorithm would reach the optimal solution, however, applying the stopping criterion gives a good quality solution in reasonable time. 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.

Acknowledgements

We thank Richard Standing, from R^2 Data Labs (Rolls-Royce), for his guidance and suggestions. We gratefully acknowledge the support of the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training and Rolls-Royce.

References

Ackert, S., April 2011. The Relationship between an Aircraft's Value and its Maintenance Status. Tech. rep., Aircraft Monitor.

Aircraft Analysis & Fleet Planning, February/March 2005. Can the 787 & A350 transform the economics of long-haul services? Tech. rep., Aircraft commerce.

Allamraju, H., 2014. Flighradar24 Unofficial API: `pyflightdata`. `http://pyflightdata.readthedocs.io/en/latest/about.html`, accessed: 17/10/2018.

Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., Shenoi, R. G., Mar. 1998. Flight String Models for Aircraft Fleeting and Routing. Transportation Science 32 (3), 208–220.

Clarke, L., Johnson, E., Nemhauser, G., Zhu, Z., Jan 1997. The aircraft rotation problem. Annals of Operations Research 69 (0), 33–46.

Cohn, A. M., Barnhart, C., 2003. Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions. Operations Research 51 (3), 387–396.

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.

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.

Department for BIS, February 2016. UK Aerospace Maintenance, Repair, Overhaul & Logistics Industry Analysis. BIS Rearch Paper Number 275.

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

Feo, T. A., Bard, J. F., 1989. Flight scheduling and maintenance base planning. Management Science 35 (12), 1415–1432.

Flightradar24 AB, 2018. Flightradar24. `www.flightradar24.com`, accessed: 17/10/2018.

Gopalan, R., Talluri, K. T., 1998. The Aircraft Maintenance Routing Problem. Operations Research 46 (2), 260–271.

Gurobi Optimization version 8.0, I., 2018. Gurobi Optimizer Reference Manual. http://www.gurobi.com, accessed: 17/10/2018.
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.
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.
Haouari, M., Shao, S., Sherali, H. D., 2013. A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem. Transportation Science 47 (4), 508–525.
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.
Khaled, O., Minoux, M., Mousseau, V., Michel, S., Ceugniet, X., 2018. A Compact Otimization Model for the Tail Assignment Problem. European Journal of Operational Research 264 (2), 548–557.
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.
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.
Levin, A., 1971. Scheduling and Fleet Routing Models for Transportation Systems. Transportation Science 5 (3), 232–255.
Li, Z., Guo, J., Zhou, R., Jan 2016. Maintenance scheduling optimization based on reliability and prognostics information. In: Annual Reliability and Maintainability Symposium (RAMS). pp. 1–5.
Liang, Z., Chaovalitwongse, W., 11 2013. A Network-Based Model for the Integrated Weekly Aircraft Maintenance Routing and Fleet Assignment Problem. Transportation Science 47, 493–507.
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.
Naber, A., 2017. Resource-Constrained Project Scheduling with Flexible Resource Profiles in Continuous Time. Computers & Operations Research 84, 33–45.
Nocedal, J., Wright, S., 2006. Numerical Optimization: Springer Series in Operations Research and Financial Engineering. Springer.
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.
Papadakos, N., 2009. Integrated airline scheduling. Computers and Operations Research 36 (1), 176–195.
Qantas, July 2016. Qantas News Room: The A, C and D of Aircraft Maintenance. www.qantasnewsroom.com.au/roo-tales/the-a-c-and-d-of-aircraft-maintenance/, accessed: 17/10/2018.
Safaei, N., Jardine, A. K., 2018. Aircraft routing with generalized maintenance constraints. Omega 80, 111 – 122.
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.
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.
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.
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.