Les Cahiers du GERAD ISSN: 0711–2440

The tail assignment problem with look-ahead maintenance constraints

S.J. Maher, G. Desaulniers,
F. Soumis

G–2015–127

December 2015

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2015.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2015.

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

The tail assignment problem with look-ahead maintenance constraints

Stephen J. Maher [a]

Guy Desaulniers [b]

François Soumis [b]

[a] *Department of Optimization, Zuse Institute Berlin, 14195 Berlin, Germany*

[b] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

`maher@zib.de`
`guy.desaulniers@gerad.ca`
`francois.soumis@gerad.ca`

December 2015

Les Cahiers du GERAD
G–2015–127

Abstract: The tail assignment problem is a critical part of the airline planning process that assigns specific aircraft to sequences of flights, called lines-of-flight, to be operated the next day. The aim of this paper is to develop an operationally flexible tail assignment that satisfies short-range—within the next three days—aircraft maintenance requirements and performs the aircraft/flight gate assignment for each input line-of-flight. While maintenance plans commonly span multiple days, the related tail assignment problems can be overly complex and provide little recourse in the event of schedule perturbations. The presented approach addresses operational uncertainty by extending the one-day routes aircraft maintenance routing approach to satisfy maintenance requirements explicitly for the current day and implicitly for the subsequent two days. A mathematical model is presented that integrates the gate assignment and maintenance planning problems. To increase the satisfaction of maintenance requirements, an iterative algorithm is developed that modifies the fixed lines-of-flight provided as input to the tail assignment problem. The tail assignment problem and iterative algorithm are demonstrated to effectively satisfy maintenance requirements within appropriate run times using input data collected from three different airlines.

Key Words: Tail assignment, column generation, iterative algorithm.

1 Introduction

The tail assignment problem (TAP) is a component of the airline planning process—separated by aircraft type—that involves the assignment of tasks to aircraft to satisfy operational constraints. A task within the TAP is a sequence of flights that can be performed by a single aircraft, which is termed a line-of-flight (LOF). Operational constraints include: ensuring each flight is operated by an aircraft, satisfying aircraft maintenance requirements and planning through flights. Traditionally, both the LOFs and the operational constraints are provided as input. Hence, there is a strong correlation between the construction of LOFs and the ability for planners to satisfy operational constraints from the solution to the TAP.

The LOFs provided as input for the TAP are generated across a number of stages of the airline planning process. The aircraft routing problem (ARP) is part of the airline planning process that is critical in the LOFs construction. As a result, the structure of the LOFs is dependent on the formulation of the ARP, which directly impacts the formulation of the TAP. A common consideration of the ARP is the planning of aircraft maintenance opportunities. In a review of the ARP, Lacasse-Guay et al. [11] state that the different formulations can be categorised by the method of LOF construction—*string*, *big-cycle* and *one-day routes* approaches. The string and big-cycle approaches are commonly solved over a number of days to provide a maintenance plan for the complete airline fleet. Alternatively, the one-day routes approach is solved to identify LOFs spanning a single day for all aircraft. Maintenance planning is performed by only considering the aircraft requiring maintenance at the end of the current day. It is important to note that all of these approaches can be solved many months in advance of the day of operations.

The string maintenance planning approach constructs a set of generic flight sequences, each to be performed by a single aircraft, that originate and terminate at maintenance opportunities. An individual flight sequence, termed a flight route, is constructed to be maintenance feasible. As such, the solution to a set partitioning problem, selecting a set of flight routes that covers all flights within a given time period, satisfies a fleet's maintenance requirements. Examples of the string maintenance planning approach are presented by Barnhart et al. [2] and Sriram and Haghani [16]. The big-cycle approach to maintenance planning involves the construction of a single route spanning multiple days that covers every flight in the schedule. Equal utilisation motivates this approach. This is achieved by constructing a single cycle that includes all flights to be operated by all aircraft. Examples of the big-cycle approach are presented by Feo and Bard [6], Clarke et al. [4] and Gopalan and Talluri [7]. It is important to note that Feo and Bard [6] and Gopalan and Talluri [7] construct big-cycle solutions using sets of LOFs that span a single day. Finally, the one-day routes approach is vastly different from the two previously discussed. This approach is solved to identify flight routes that span a single day. The objective of this approach is to ensure that a sufficient number of flight routes from each airport terminate at a maintenance station so that the maintenance critical aircraft can receive maintenance that night. The one-day routes ARP is inherently stochastic since it assumes operations from previous days will perturb the maintenance plan. Examples of the one-day routes approach are presented by Heinhold [9], Lapp and Cohn [12] and Maher et al. [13].

While the ARP approaches presented above involve the generation of LOFs for input to the TAP, alternative methods have been proposed that combine the ARP and TAP. One of the most detailed investigations of the TAP developed in this manner is presented by Grönkvist [8]. The TAP proposed by [8] is solved a month at a time and comprises features from the fleet assignment, aircraft routing, maintenance planning and through assignment (matching high valued in-bound and out-bound flights with the same aircraft) problems. The construction of LOFs within the TAP to minimise a robustness measure is presented by Borndörfer et al. [3]. In addition, [3] construct LOFs to adjust for any perturbations from preceding days that may have affected the maintenance plan. Finally, flexibility in the construction of LOFs is achieved by Ruther [14] by considering the TAP as a component of an integrated airline planning problem. The problem is posed to be solved approximately four days before the day of operations to adjust the planned solutions in response to schedule perturbations.

There is a growing interest in methods to handle operational uncertainty while satisfying maintenance requirements. The one-day routes approach addresses operational uncertainty by planning LOFs to span only a single day. Tail assignment problems handle schedule perturbations by reconstructing LOFs. The problem presented in this paper incorporates aspects from both approaches. The TAP developed in this paper is solved using one-day routes as input and satisfies day-one maintenance requirements by permitting the reconstruction of only a subset of LOFs. The considered problem builds on the one-day routes approach with the development of a suitable method to assign LOFs constructed within this framework. As an extension to both the one-day routes approach and TAP, day-two and day-three maintenance requirements are implicitly satisfied using look-ahead constraints. Finally, a critical consideration arising from the use of one-day routes as input—since the routes only span a single day—is the over the night gate assignments. The developed problem presents a method integrating the assignment of LOFs with the matching of aircraft locations to the originally planned gate assignments.

Various solution approaches have been applied to solve the tail assignment and maintenance planning problems. Column generation is popular for problem formulations where aircraft routes are not provided a priori. This is a feature of the TAP developed by Grönkvist [8], Borndörfer et al. [3] and Ruther [14] and the maintenance planning approaches by Barnhart et al. [2] and Maher et al. [13]. There are many cases where the LOFs are generated by an ARP and provided as input to the TAP. For such problem formulations, solution methods including Lagrangian relaxation and subgradient approaches [4], problem specific heuristics [6, 7, 16] or general-purpose mixed-integer programming solvers [12] have been employed. Observations suggest that exact solution approaches, such as column generation, can be overly time-consuming and not suitable for the practical implementation of algorithms. However, the high solution quality that is achievable using exact solution approaches is desired.

A compromise between the exact solution approach of column generation and problem-specific heuristics is presented in the form of iterative solution algorithms [18, 5]. Iterative algorithms have previously been employed to solve integrated airline planning problems, whereby the solution to one stage can be fixed prior to solving the alternate stage and then iterating between the two problems. Extending the development of this technique an iterative solution algorithm is presented to improve the solution of the TAP using a set of input LOFs. The algorithm involves i) solving the TAP to identify any infeasibilities in the maintenance plan, and ii) solving approximately a relaxation of the TAP to generate flight routes for a subset of aircraft to address these infeasibilities. The algorithm executes in a run time less than that required by exact approaches while still achieving high-quality solutions. The iterative algorithm is a contribution of this paper.

The TAP considering maintenance requirements and gate assignment changes is developed in this paper. The contributions of this paper are i) the consideration of over-the-night gate assignments within the TAP, ii) the development of a TAP using one-day routes as input, iii) the implicit consideration of day-two and day-three maintenance requirements using look-ahead constraints, and iv) the novel iterative algorithm that improves the maintenance planning achieved by the TAP. The problem description and formulation is presented in Section 2. The discussion in this section will involve two parts, the first presenting the TAP to satisfy gate assignments and maintenance requirements for day one and the second introducing the look-ahead maintenance constraints. An iterative algorithm is developed in Section 3 that aims to improve the solution to the TAP when solved using a set of input LOFs. Section 4 will describe the data used to evaluate the TAP developed in this paper. The computational results involving various flight schedules will be presented in Section 5. Finally, Section 6 will provide some concluding comments.

2 The tail assignment problem

The TAP is solved immediately prior to the day of operations to aid the recovery of planned assignments that are disrupted as a result of schedule perturbations. Three critical features of aircraft routing that are highly susceptible to schedule perturbations are addressed by the TAP: the assignment of LOFs to aircraft, maintenance requirements and over-the-night gate assignments. The TAP is solved at an arbitrary time when it is expected that most aircraft are located on the ground. The fleets considered for the TAP are short-haul and medium-haul fleets, hence there are no overnight flights. However, airlines may operate in multiple time zones. Thus, all flights departing between midnight and midnight (local time) on consecutive days are considered to belong to the same one-day schedule. It is possible for aircraft to be operating flights while the TAP is being solved, requiring an estimated arrival time to be used in this problem for such aircraft. Finally, to satisfy regulatory requirements aircraft are expected to receive maintenance once every six days.

This paper presents the TAP defined by: Given a set of input LOFs that span the next day of operation, assign to each aircraft exactly one LOF that originates from the current aircraft location. The LOF assignment minimises the number of day-one maintenance-critical aircraft *not* terminating at a maintenance base at the end of the next day. Additionally, the assignment of LOFs to aircraft will minimise the cost associated with any required gate assignment changes. A further aspect unique to this paper is the consideration of day-two and day-three maintenance requirements. Two additional sets of LOFs—one each for days two and three—that span a single day are provided as input. All three sets of input LOFs are not required to be identical. The number of available maintenance routes from the end of day one to the end of days two and three are computed from the day-two and day-three LOFs. Using the number of available maintenance routes as input, the TAP minimises the number of day-two and day-three maintenance critical aircraft unable to receive maintenance on the respective days without the explicit assignment of LOFs.

Explicit maintenance planning is only modelled in the TAP for the forthcoming day of operation. However, aircraft requiring maintenance on days two and three are still considered. This is supported by the addition of constraints that ensure a sufficient number of maintenance routes depart from each overnight airport on day two and day three for the maintenance critical aircraft. This implicit consideration of maintenance requirements is a novel approach that has not been previously investigated. For ease of exposition Section 2.1 presents the TAP that only considers the day-one maintenance planning. Additionally in Section 2.1, the over-the-night assignment of aircraft to gates is presented—a novel feature of the TAP. Finally, an approach to implicitly enforce the day-two and day-three maintenance requirements using look-ahead maintenance constraints is presented in Section 2.2.

2.1 The tail assignment problem without look-ahead constraints

There exist four key components of the TAP, namely the overnight airports, aircraft, available gates and the LOFs. Let B be the set of overnight airports. A set of aircraft R^b, all of the same type, are located at each overnight airport $b \in B$ to commence the forthcoming day of operation. The LOFs available for assignment to an aircraft $r \in R^b$ are given by the set P^r, which is indexed by p. P^r is populated with LOFs that are identical for all $r \in R^b$ originating for the same overnight airport b. Finally, the set of all available gates at airport b is given by G^b.

There are three problems presented in the description of the TAP: the assignment of LOFs, maintenance planning and gate assignment. The LOF assignment is modelled using the binary variables y^r_p that equal one if aircraft r is assigned to LOF p, and zero otherwise. The origination and termination locations of the LOF and the contained flights are directly considered in the TAP model. All flights of an airline schedule, which belong to the set denoted by N, are each included in exactly one input LOF. The parameter a_{fp} is defined to equal one if flight f is included in LOF p. Each LOF p originates and terminates at an overnight airport. A subset of overnight airports, $\hat{B} \subset B$, are identified as maintenance stations. In regards to the second problem, the day-one maintenance requirements are addressed by attempting to assign each maintenance critical aircraft to an LOF terminating at a maintenance station. The parameters o_p are defined to equal one if the LOF p terminates at a maintenance station, and zero otherwise. A given aircraft $r \in R^b, b \in B$ is identified as maintenance critical, defined as requiring maintenance at the end of the current day, by the parameter $\theta^r_1 = 1$. Since disruptions from preceding days may prohibit aircraft from entering a maintenance station, the slack variables s^r_1, $r \in R^b$, $b \in B$, are introduced, with the objective coefficient c^r_1, to penalise any infeasibility of the maintenance plan.

The third problem of the TAP is the assignment of gates so that aircraft can operate the assigned LOFs. The gates at an airport are a scarce resource that are required by various flights and aircraft across multiple time periods during the day. Since a flight only requires a gate for the time immediately prior to departure, this problem can be modelled using a time discretisation. A time period lies between the departure of two consecutive flights at a given airport. The set of all time periods during which a flight f may use and occupy a gate is given by T^f—containing only a single time period for the current application.

Since the gate assignment problem involves three different resources—LOFs, aircraft and gates—there are various decisions that must be considered. Figure 1 displays an example of the possible gate reassignments for a single LOF and aircraft. The light grey node represents the originally planned gate assignment for the LOF, given by $g_F(p)$ for LOF p. Since an LOF does not occupy any physical space this node can be treated as a dummy node in the graph. The dark grey nodes represent the current gate location for an aircraft, given by $g_R(r)$ for aircraft r. Multiple dark grey nodes are required in the graph to model the parking of an aircraft across time periods. The white nodes represent the available gates in each time period.

All possible gate assignment decisions for an LOF/aircraft pairing are represented by the directed edges in Figure 1. The dotted edges between the light grey and white nodes are the assignment of an LOF to a gate for departure. Since T^f, $f \in N$, only contains a single time period t, edges only exist between the light grey node and the white nodes of period t. The dashed edges between the dark grey and white nodes represent the movement of an aircraft to a gate for departure—occurring in any time period. Finally, the solid edges between the dark grey nodes identify the parking of an aircraft at its current gate location. A feasible solution to this problem requires the selection of exactly one dotted and one dashed edge—indicated by the bold edges in the graph. The selected dotted and dashed edges must be incident to the same white node. Edges between the dark grey nodes must be selected such that flow is maintained from the first dark grey node to a white node. The resulting edge selection represents the parking and movement of aircraft.

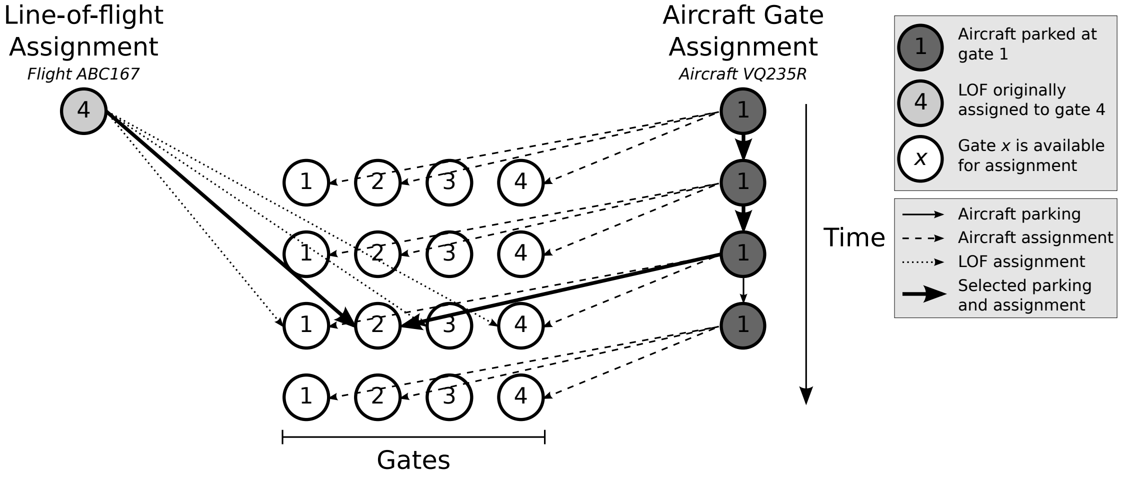


Figure 1: Example graph formulation of the gate assignment problem. The central nodes (white) represent final gate assignment to be implemented. The light grey node represents a single LOF original assignment and the dark grey nodes represent a single aircraft location.

As demonstrated in Figure 1, two different gate reassignments must occur, the LOF and the aircraft. To facilitate the LOF/aircraft gate matching the set $\hat{N}^b$ is defined to contain all initial flights of LOFs departing from overnight airport b. Following the departure of each flight a small time window, known as the gate turnaround time, is required to prepare the gate for the next flight departure. The set Q^f, termed a conflict set, is thus defined to contain all flights that depart within the gate turnaround time of flight f. The conflict set is used to ensure at most one flight is assigned to a gate in each time period. All departure time periods for overnight airport b is given by $T^b = \cup_{f \in \hat{N}^b} T^f$. To aid the problem formulation, the final time period at overnight airport b is given by $\bar{T}^b$.

Gate assignment changes are modelled using the set of binary variables x^r_{ikt} that equal one if the LOF originally assigned to gate i now originates from gate k during time period t to be operated by aircraft r. These variables represent the selection of the dotted edges in Figure 1. The cost associated with changing the LOF gate assignment at airport $b \in B$ from gate $i \in G^b$ to $j \in G^b$ is given by c_{ij}.

The aircraft gate assignment variables are separated into two sets to indicate the parking of an aircraft and the gate reassignment. Both sets of variables are defined with a time reference for both the current aircraft location and the final gate assignment to model the occupation of gates. The reassignment variables

$\hat{x}^r_{kjt-1}$ equal one if aircraft r, which is occupying gate j in time period $t-1$, is moved to gate k in time period t: represented by the selection of a dashed edges in Figure 1. The aircraft parking variables $\bar{x}^r_{jt}$ equal one to indicate that aircraft r occupies gate j from the start of time period t to the start of $t+1$. This is represented by the selection of an edge between the dark grey nodes in Figure 1. Once moved, an aircraft is no longer able to be "parked" at a gate and must depart during that time period. The cost of moving an aircraft from gate i to j is given by c'_{ij}.

The TAP can be modelled as the following mixed integer program:

$$\min \quad \sum_{b\in B}\sum_{r\in R^b} c^r_1 s^r_1 + \sum_{b\in B}\sum_{r\in R^b}\sum_{i\in G^b}\sum_{j\in G^b}\sum_{t\in T^b}\left\{c_{ij}x^r_{ijt} + c'_{ij}\hat{x}^r_{ijt-1}\right\}, \tag{1}$$

$$\text{s.t.} \quad \sum_{b\in B}\sum_{r\in R^b}\sum_{p\in P^r} a_{fp}y^r_p = 1 \quad \forall f \in N, \tag{2}$$

$$\sum_{p\in P^r} y^r_p \leq 1 \quad \forall b \in B, \forall r \in R^b, \tag{3}$$

$$\sum_{p\in P^r} o_p y^r_p + s^r_1 \geq \theta^r_1 \quad \forall b \in B, \forall r \in R^b, \tag{4}$$

$$\sum_{t\in T^f}\sum_{k\in G^b} x^r_{ikt} - \sum_{p\in P^r} a_{fp}y^r_p = 0 \quad \forall b \in B, \forall r \in R^b, \forall f \in \hat{N}^b, i = g_F(f), \tag{5}$$

$$\sum_{r\in R^b}\ \sum_{\bar{f}\in Q^f\cup\{f\}}\ \sum_{\substack{i\in G^b|\\ i=g_F(\bar{f})}}\ \sum_{t\in T^{\bar{f}}} x^r_{ikt} \leq 1 \quad \forall b \in B, \forall f \in \hat{N}^b, \forall k \in G^b, \tag{6}$$

$$\sum_{t\in T^b}\sum_{k\in G^b} \hat{x}^r_{kjt-1} = 1 \quad \forall b \in B, \forall r \in R^b, j = g_R(r), \tag{7}$$

$$\sum_{i\in G^b} x^r_{ikt} - \hat{x}^r_{kjt-1} = 0 \quad \forall b \in B, \forall r \in R^b, j = g_R(r), \forall k \in G^b, \forall t \in T^b, \tag{8}$$

$$\bar{x}^r_{jt-1} - \left(\bar{x}^r_{jt} + \sum_{k\in G^b}\hat{x}^r_{kjt}\right) = 0 \quad \forall b \in B, \forall r \in R^b, j = g_R(r), \forall t \in T^b\backslash\bar{T}^b, \tag{9}$$

$$\sum_{\substack{r\in R^b|\\ j=g_{AIR(r)}}} \hat{x}^r_{kjt-1} + \sum_{\substack{r\in R^b|\\ k=g_R(r)}} \bar{x}^r_{kt-1} \leq 1 \quad \forall b \in B, \forall k \in G^b, \forall t \in T^b\backslash\bar{T}^b, \tag{10}$$

$$\sum_{\substack{r\in R^b|\\ j=g_{AIR(r)}}} \hat{x}^r_{kjt-1} \leq 1 \quad \forall b \in B, \forall k \in G^b, t = \bar{T}^b, \tag{11}$$

$$y^r_p \in \{0,1\} \quad \forall r \in R, \forall p \in P, \tag{12}$$

$$s^r_1 \geq 0 \quad \forall r \in R, \tag{13}$$

$$x^r_{ijt} \in \{0,1\} \quad \forall b \in B, \forall r \in R^b, \forall i,j \in G^b, \forall t \in T^b, \tag{14}$$

$$\hat{x}^r_{ijt-1} \in \{0,1\} \quad \forall b \in B, \forall r \in R^b, \forall i \in G^b, j = g_R(r), \forall t \in T^b, \tag{15}$$

$$\bar{x}^r_{jt-1} \in \{0,1\} \quad \forall b \in B, \forall r \in R^b, j = g_R(r), \forall t \in T^b\backslash\bar{T}^b. \tag{16}$$

The objective of the TAP minimises the violation of maintenance requirements and the cost associated with any required gate assignment changes. While it is possible to formulate the tail assignment as a set partitioning of input LOFs, it is more convenient to formulate this problem as a set partitioning of flights in the network. The latter formulation aids the development of a re-optimisation method that modifies LOFs when (1)–(16) is maintenance infeasible, which is presented in Section 3. As such, constraints (2) ensure that every flight $f \in N$ is assigned to an aircraft. Each aircraft must operate exactly one LOF, which is given by constraints (3). The day-one maintenance requirements for the maintenance critical aircraft are enforced with constraints (4). These constraints include a slack variable s^r_1 to penalise any maintenance violations.

The gate assignment problem is given by constraints (5)–(11) and (14)–(16). Two different operations are necessary for aircraft r to operate LOF p: the reassignment of LOF gate i to k and the movement of the aircraft from j to k. However, it is possible that $i = k$, $j = k$, or both. The first operation is modelled by identifying the LOF gate assignment changes using constraints (5): selecting a dotted edge on the left-hand side of Figure 1. To avoid any conflicts related to flight departures at gates, at most one flight within each conflict set may use a gate in a given time window, which is expressed by constraints (6). The second gate assignment operation, the aircraft movement, is imposed by constraints (7): selecting a dashed edge from the right-hand side of Figure 1. The flow balance constraints (8) are provided to ensure that the LOF and aircraft are set to have the same departure gate within the same time period. The dynamics of aircraft at gates, either parking or reassignment, is given by the flow balance constraints (9). Specifically, for every incoming aircraft parking edge there must exist either an outgoing parking edge or a reassignment edge. Finally, it is only possible for each gate in each time period to be assigned to only one aircraft/LOF pair or be occupied by a parked aircraft. This condition is given by constraints (10)–(11).

2.2 Modelling the look-ahead maintenance constraints

A limitation of the one-day routes approach is that routing information is only available for the current day of operation. As such, it is difficult to explicitly plan maintenance on days two or three. However, the solution to the TAP provides aircraft terminating locations at the end of day one. This information can be effectively used in look-ahead maintenance constraints to implicitly satisfy day-two and day-three maintenance requirements.

Satisfying the maintenance requirements at the end of days two and three is achieved by comparing the number of maintenance critical aircraft and the number of available maintenance routes from each airport. The set of maintenance routes is given by the fixed LOFs; however, only the origination and termination locations are considered in the look-ahead maintenance constraints. Since the LOFs need not be identical on each day, the set of termination locations for a given overnight base may differ between days. Figure 2 presents an example focusing on one airport, labelled as SYD, displaying the total number of departing LOFs and the available routes to satisfy the day-two and day-three maintenance requirements. This figure shows that the maintenance routes are a subset of the available LOFs. From a total of seven LOFs on day two, there are only three that terminate at a maintenance station, shown in Figure 2b. To satisfy the day-two maintenance requirements, all maintenance critical aircraft in SYD at the end of day one must be assigned to one of these routes on day two. Figure 2c shows the available routes terminating at a maintenance station at the end of day three. Comparing Figures 2a and c, on day three there are a total of seven LOFs from all airports terminating at a maintenance station but only six day-three maintenance routes are available for aircraft located at SYD at the end of day one. This is due to two bottlenecks in the number of maintenance routes: from BNE on day three there is only one LOF terminating at a maintenance station and there are only two LOFs from SYD terminating at SYD at the end of day two. The ability to perform maintenance checks on maintenance critical aircraft on day three depends on the number of LOFs on day two and the number of maintenance LOFs on day three.

The day-two maintenance requirements are satisfied by simply counting the number of maintenance critical aircraft terminating at each airport at the end of day one. For simplicity the notation b_i is used to identify airport b at the start of day i. The parameter θ_2^r equals one if aircraft r requires maintenance at the end

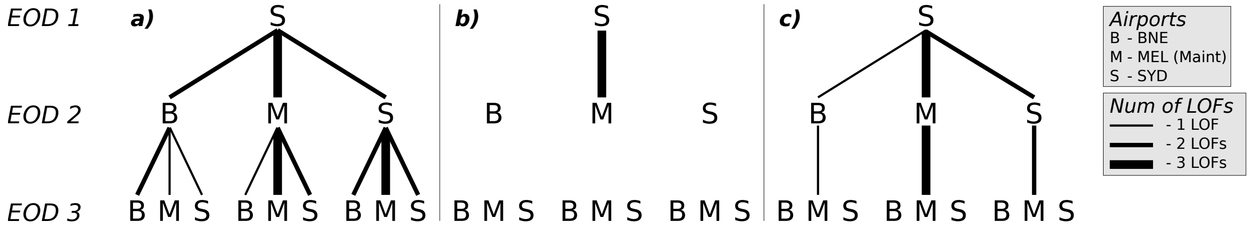


Figure 2: Available LOFs and maintenance routes from SYD at the end of day 1. a) All available LOFs. b) Routes from SYD terminating at the maintenance station (MEL) at end of day two. c) Routes from SYD terminating at the maintenance station (MEL) at the end of day three.

of day two, and zero otherwise. Using this parameter and the LOF assignment it is possible to count the number of maintenance critical aircraft located at each airport at the start of day two. The total number of maintenance routes departing from airport b on day two is given by the parameter M^2_b, which is computed using the fixed LOFs for day-two. The difference between the number of day-two maintenance critical aircraft and M^2_b indicates the feasibility of the day-one tail assignment. To satisfy day-two maintenance requirements any over demand for maintenance routes, indicating a maintenance plan infeasibility, must be penalised in the objective function. This is achieved through the addition of a set of slack variables $s^{b_2}_2$, $b_2 \in \hat{B}$. The location of the aircraft at the end of day one is determined by the LOF assignment. The termination location of LOF p is given by the parameter $term(p)$. The addition of the following constraints evaluate the satisfaction of maintenance requirements for the day-two maintenance critical aircraft:

$$\sum_{b_1 \in B} \sum_{r \in R^{b_1}} \sum_{\substack{p \in P^{b_1} | \\ term(p) = b_2}} \theta^r_2 y^r_p - s^{b_2}_2 \leq M^2_{b_2} \quad \forall b_2 \in B, \tag{17}$$

$$s^{b_2}_2 \geq 0 \quad \forall b_2 \in B. \tag{18}$$

Constraints (17) count the number of day-two maintenance critical aircraft located at b_2 and sets the slack variable $s^{b_2}_2$ to penalise any maintenance misalignments.

Satisfying the day-three maintenance requirements involves counting the number of day-three maintenance critical aircraft and assigning a day-two termination location to each. The parameters θ^r_3 equal one to identify whether aircraft r requires maintenance at the end of day three, and zero otherwise. The number of paths originating from airport b_2 that include overnight airport b_3 and terminate at a maintenance station at the end of day three is given by the parameter $N^{b_2 b_3}$. The variables $\xi_{b_2 b_3}$ count the number of day-three maintenance critical aircraft located at b_2 arriving at airport b_3 at the end of day two. Similar to days one and two, the number of maintenance routes departing from overnight airport b_3 is given by the parameter $M^3_{b_3}$, which is computed using the fixed LOFs for day-three. The additional constraints to evaluate maintenance feasibility for the day-three maintenance critical aircraft are given by

$$\sum_{b_1 \in B} \sum_{r \in R^{b_1}} \sum_{\substack{p \in P^{b_1} | \\ term(p) = b_2}} \theta^r_3 y^r_p - s^{b_2}_3 = \sum_{b_3 \in B} \xi_{b_2 b_3} \quad \forall b_2 \in B, \tag{19}$$

$$\sum_{b_3 \in \hat{B}} \xi_{b_2 b_3} + \sum_{b_1 \in B} \sum_{r \in R^{b_1}} \sum_{\substack{p \in P^{b_1} | \\ term(p) = b_2}} \theta^r_2 y^r_p - s^{b_2}_2 \leq M^2_{b_2} \quad \forall b_2 \in B, \tag{20}$$

$$\sum_{b_2 \in B} \xi_{b_2 b_3} \leq M^3_{b_3} \quad \forall b_3 \in B, \tag{21}$$

$$\xi_{b_2 b_3} \in [0, N^{b_2 b_3}] \quad \forall b_2, b_3 \in B, \tag{22}$$

$$s^{b_2}_3 \geq 0 \quad \forall b_2 \in B. \tag{23}$$

Constraints (19) count the number of day-three maintenance critical aircraft located at b_2 at the end of day one. This constraint also assigns each of the maintenance critical aircraft to a maintenance route from b_2 passing through b_3 using the variables $\xi_{b_2 b_3}$. Since the LOFs are provided as input, it is possible that the number of day-three maintenance critical aircraft arriving at b_2 is greater than $N^{b_2 b_3}$—causing an infeasibility. The slack variable $s^{b_2}_3$ is introduced to measure the extent of this infeasibility, which is penalised in the objective function. Setting $\xi_{b_2 b_3}$ by constraint (19) determines the number of maintenance critical aircraft originating from b_3 at the start of day three. Since day-two maintenance critical aircraft require a maintenance route departing from b_2, this reduces the number of maintenance routes passing through b_3 that are available for day-three maintenance critical aircraft, where $b_3 \in \hat{B}$. Hence, (20) constrain the number of day-two and day-three maintenance critical aircraft terminating at maintenance bases at the end of day two to at most $M^2_{b_2}$. To ensure that at most $N^{b_2 b_3}$ day-three maintenance critical aircraft arrive at b_3 from b_2 an upper bound is imposed on the variables $\xi_{b_2 b_3}$ as indicated by constraints (22).

Including the day-two and day-three look-ahead maintenance constraints introduces dominated inequalities. Specifically, constraints (17) are completely dominated by constraints (20). As such, only constraints (20) are required in the implementation of the TAP with day-two and day-three maintenance look-ahead.

The addition of constraints (18) and (19)–(23) to the TAP requires the modification of the objective function (1). This modification involves adding for each overnight airport $b \in B$ the slack variables s_2^b and s_3^b, which count the number of day-two and day-three maintenance misalignments, along with the cost parameters c_2^b and c_3^b respectively. The objective function used for the TAP with look-ahead maintenance constraints is given by

$$\sum_{b\in B}\sum_{r\in R^b} c_1^r s_1^r + \sum_{b\in B}\left\{c_2^b s_2^b + c_3^b s_3^b\right\} + \sum_{b\in B}\sum_{r\in R^b}\sum_{i\in G^b}\sum_{j\in G^b}\sum_{t\in T^b}\left\{c_{ij}x_{ij}^r + c'_{ij}\hat{x}_{ij}^r\right\}. \qquad (24)$$

3 Solution approach

The TAP is solved using a fixed set of LOFs that are constructed using the one-day routes approach. This problem can be directly solved using general purpose solvers, such as CPLEX [10] or SCIP [1]. However, by using fixed LOFs it may not be possible to satisfy maintenance requirements for all realisations of aircraft locations and maintenance plans. In particular, schedule perturbations may render the constructed LOFs infeasible for satisfying maintenance requirements. A route adjustment process that modifies the set of LOFs may be necessary to address the impact of schedule perturbations on the maintenance planning solution.

The LOF adjustment process is an iterative algorithm that involves *evaluation* and *update* stages. The *evaluation* stage determines whether the maintenance plan can be satisfied with a fixed set of LOFs by directly solving the TAP—using a general purpose solver. The *update* stage identifies alternative day-one LOFs for a subset of aircraft. The new set of LOFs is constructed with the aim to reduce maintenance misalignments on all three days. However, only the set of day-one LOFs is modified, leaving the LOFs for days two and three unchanged. It is not necessary to modify the day-two and day-three LOFs, since they will be modified as day-one LOFs when the TAP is solved on subsequent days. The update stage employs a column generation approach. The developed iterative algorithm is a compromise between fixed and fully flexible methods for solving the TAP.

The features of the evaluation stage and the update process of the set of LOFs is presented in the following sections. Section 3.1 describes the method employed to identify aircraft included in the route adjustment process of the update stage. The route adjustment problem solved in the update stage is presented in Section 3.2. The update of LOFs for the TAP is explained in Section 3.3. Finally, the termination criteria for the iterative algorithm is presented in Section 3.4.

3.1 Identify aircraft for route adjustment

The solution to the evaluation stage is used to identify a subset of aircraft that require the generation of LOFs in the update stage. The aircraft selection is based upon maintenance requirements and the assignment of LOFs in the evaluation stage. The set of selected aircraft is given by $\hat{R}$. Only a subset of flights are used to generate LOFs in the route adjustment problem. This subset of flights, denoted by $\hat{N}$, is given by those appearing in LOFs assigned in the evaluation stage to aircraft in $\hat{R}$. Throughout the algorithm, updates to $\hat{R}$ induce updates to $\hat{N}$.

Following the first execution of the evaluation stage two different types of aircraft are identified for inclusion in $\hat{R}$. The first are day-one maintenance critical aircraft assigned to an LOF that does not terminate at a maintenance station. The second are the aircraft assigned LOFs terminating at maintenance stations that do not require maintenance at the end of day one.

Subsequent iterations of the algorithm augment $\hat{R}$ using the solution to the evaluation stage. The augmentation involves identifying intersecting LOFs. Two LOFs intersect if there exists a flight in one LOF that departs within an intersection window commencing after the arrival of a flight in the other LOF at the same airport. The intersection window has a duration that is given by the sum of the minimum time aircraft require between two connecting flights, called the turn time, and a small buffer. Aircraft are selected for inclusion in $\hat{R}$ if they are assigned LOFs intersecting with LOFs assigned to aircraft in $\hat{R}$.

3.2 Route adjustment problem

The main focus of the route adjustment problem is to identify day-one LOFs for aircraft in $\hat{R}$ that minimise the maintenance misalignments for the whole fleet. As such, the gate assignment component of the TAP is not important. Consequently, the route adjustment problem corresponds to a relaxation of the TAP that is formulated with only a subset of constraints of the TAP as follows:

$$\min\left\{\sum_{b\in B}\sum_{r\in R^b} c_1^r s_1^r + \sum_{b\in B}\left\{c_2^b s_2^b + c_3^b s_3^b\right\} : (2)\text{–}(4),\ (12)\text{–}(13),\ (17)\text{–}(23)\right\}. \tag{25}$$

Model (25) does not require any modification to the constraints defined for the TAP, which are implemented as presented in Section 2.

Note that all feasible tail assignments obtained by solving (25) are also feasible for the TAP. Indeed, it is always possible to determine a feasible gate assignment for a feasible tail assignment. However, computing an optimal gate assignment for a fixed tail assignment might result in a suboptimal solution to the TAP.

The LOFs provided as input for the TAP are used to define the initial set of variables in model (25). The variables, or columns, defined by the input LOFs are a subset of all possible aircraft routes for the given flight schedule. Hence, formulating this problem using these variables alone represents a restriction of the full problem formulation. Model (25) can then be described as a restricted master problem (RMP) for the column generation solution approach. A column generation algorithm is applied to generate routes for the RMP to reduce the maintenance misalignments identified by the solution to the TAP.

A column generation subproblem is formed for each aircraft $r \in R^b \cap \hat{R}, b \in B$. The objective of each subproblem is to identify the aircraft routing variable with the minimum reduced cost. To facilitate the description of the column generation subproblem, the dual variables related to the constraints of (25) that appear in aircraft route variables reduced cost function will be presented. For ease of exposition, the constraints will be numbered with respect to their initial presentation in Section 2. The dual variables for the flight coverage constraints (2) are defined as $\boldsymbol{\rho} = \{\rho_j, \forall j \in N\}$. For the LOF assignment constraints (3), the dual variables are defined as $\boldsymbol{\delta} = \{\delta_b^r, \forall b \in B, \forall r \in R^b\}$. The dual variables for the day-one maintenance enforcement constraints (4) are defined as $\boldsymbol{\alpha} = \{\alpha_b^r, \forall b \in B, \forall r \in R^b\}$. For the day-two maintenance enforcement constraints (20), the dual variables are defined as $\boldsymbol{\beta} = \{\beta_b, \forall b \in B\}$. Finally, the dual variables for the day-three maintenance critical count constraints (19) are defined as $\boldsymbol{\gamma} = \{\gamma_b, \forall b \in B\}$.

The column generation subproblem for (25) is a shortest path problem: Identifying a minimum cost path through a network from a single source to one of multiple sink nodes. The network is defined by a set of nodes given by N and a set of edges given by the feasible connections between the flights contained in N. A connection between flights i and j contained in N, (i,j), is deemed feasible if i) the destination of i is the same as the origin of j, and ii) the departure time of j occurs after the minimum turn time following the arrival of i. All feasible connections are contained in the set C and the set of all connections between flights contained in $\hat{N}$ is given by $\hat{C} = \{(i,j) \in C | i \in \hat{N} \wedge j \in \hat{N}\}$. To describe the minimum cost path, the binary variables w_{ij}^r equal one to indicate aircraft r uses connection (i,j), with the objective cost c_{ij}, or zero otherwise. Aircraft must originate from an overnight airport b and may terminate at any overnight airport $b' \in B$, describing the source and sink nodes respectively. The binary parameters $\bar{o}_b$ are introduced to indicate whether maintenance can be performed at overnight airport b. Using these definitions, the column generation subproblem is given by

$$\hat{c}^r = \min \sum_{(i,j)\in\hat{C}} c_{ij} w_{ij}^r - \sum_{i\in\hat{N}\cup B}\sum_{j\in\hat{N}} \rho_j w_{ij}^r - \delta_b^r - \sum_{i\in\hat{N}}\sum_{b'\in B} w_{ib'}^r\left\{\bar{o}_{b'}\alpha_{b'}^r + \theta_2^r\beta_{b'} + \theta_3^r\gamma_{b'}\right\}, \tag{26}$$

$$\text{s.t.} \quad \sum_{i\in\hat{N}} w_{ij}^r - \sum_{k\in\hat{N}} w_{jk}^r = 0 \quad \forall j \in \hat{N}, \tag{27}$$

$$\sum_{j\in\hat{N}} w_{bj}^r = 1, \tag{28}$$

$$\sum_{j\in\hat{N}}\sum_{b'\in B} w_{jb'}^r = 1, \tag{29}$$

$$w^r_{ij} \in \{0,1\} \quad \forall (i,j) \in \hat{C}. \tag{30}$$

The objective function (26) is the reduced cost function of the routing variables for aircraft $r \in R^b \cap \hat{R}, b \in B$ in model (25). The flow balance at each node (flight) in the network is maintained by the constraints (27). The origin and destination of a flight route is enforced through constraints (28) and (29) respectively. The restrictions imposed by the set $\hat{N}$, subsequently $\hat{C}$, for the route adjustment problem reduces the problem complexity.

Branch-and-price is used to solve the route adjustment problem (25) to integer optimality. The resulting master problem solution minimises the number of maintenance misalignments given a fixed set of LOFs for aircraft $r \in R \backslash \hat{R}$. Since the gate assignment constraints have been omitted this solution may not be optimal for the TAP. Hence, the optimal columns from model (25) must be added to the TAP, which is then resolved to assess the maintenance misalignments and gate assignments.

A consideration of the route adjustment problem is the impact of alternative connections contained in the newly generated routes on crew. A feasible connection for crew requires a minimum sit time, generally longer than the minimum turn time for aircraft, between the arrival of i and the departure of j. However, crew may use the connection (i,j) with a ground time less than the minimum sit time but greater than the minimum turn time if an aircraft also uses this connection. Such connections are called *short connections*. These connections are important for the crew scheduling solution and must be protected in the generation of aircraft routes. This is achieved in the route adjustment problem by ensuring if any short connections exist in the input LOFs that these connections are used in any solution to the TAP.

3.3 Update the variables of the TAP

The solution to (25) identifies day-one LOFs that potentially reduce the maintenance misalignments in the solution to the TAP. While it is possible to add all aircraft routes generated by solving (25), observations from computational experiments indicate that this negatively impacts the efficiency of the mixed integer programming solvers. A more fractional LP solution results from the addition of all generated variables. As a consequence, the TAP requires more nodes to find the integer optimal solution. For large flight schedules, this increase in node processing can result in solution times that are impractical for the considered application. To avoid the unnecessary increase in the solution run times only the LOF variables with a positive value in the solution to (25) are selected for addition to the TAP.

3.4 Algorithm termination

The algorithm terminates when no further improvement in the maintenance misalignments can be achieved by adding variables to the TAP. This is identified using various stopping criteria. First, if the number of aircraft contained in $\hat{R}$ equals the total number of aircraft, then the algorithm is terminated. Second, an *updated* flag, which is *true* only if between iterations the TAP objective function value decreases or the number of aircraft contained in $\hat{R}$ increases, is used to identify whether the algorithm has stalled. The algorithm terminates if *updated* is *false* for two consecutive iterations. If no columns are added while solving (25) then the intersection window is increased, the set $\hat{R}$ is updated and (25) is resolved. This situation is treated as another iteration of the algorithm and if *updated* is *false* the failure count increases and the algorithm will terminate when this count is equal to three.

The route adjustment process is a heuristic approach to minimise the maintenance misalignments. As such, optimality of the TAP can not be guaranteed. However, the purpose of this algorithm is to provide a quick and efficient method to reduce the maintenance misalignments that arise from using a fixed LOF input. The computational results in Section 5 demonstrate the ability of the algorithm to achieve this task and present the difference between the identified and optimal TAP solution.

4 Model data

Inputs are required for each of the fundamental components of the TAP—the LOFs, gates and aircraft. The data required to define the TAP has been collected from previously performed research, estimates from literature or generated for this study. In regards to the data generated for this study, a number of experiments are conducted to give a broad overview of the model.

The LOFs provided as input to the TAP are constructed using the one-day routes approach and a classical aircraft routing problem. Both sets of LOFs are collected from the study by Maher et al. [13]. Two different models for generating one-day routes are presented by [13], original (SDAMRP) and recoverable robust formulations (SDAMRP-RR); however, only the routes generated from the SDAMRP formulation are used for the current computational experiments. The classical aircraft routing problem used to generate LOFs in [13] is given by a simple modification to the SDAMRP: eliminating the maintenance misalignment penalty term from the objective function. The LOFs are generated for three different flight schedules and are labelled as Fn-Am, where n is the number of flights and m is the number of aircraft. It is assumed that each airline considered is operating a cyclic schedule. As such, the LOFs are said to be repeated on each day of interest. It is trivial to relax this assumption with little change to the complexity of the problem.

Gate assignment problem must consider various operational features—the available gates, the current assignments of flights to gates and the locations of each aircraft. For this study, it was necessary to estimate the number of available gates from the input LOFs and published schedules. The estimate was made using the assumption that for a each airport the number of gates is equal to the maximum number of aircraft that are on the ground at the same time. While this may overestimate the number of gates, the formulation of the TAP permits the modelling of dummy gates. As an alternative to estimating the available gates, it may be possible to obtain the required information from publicly available sources. In particular, the gates for use at each airport for Southwest Airlines and US Airways are provided on their respective websites [15, 17].

The input LOFs generated in [13] do not include any gate assignment information. Hence, the starting gate for each LOF is arbitrarily assigned in order of departure. Similarly, the current aircraft locations are arbitrarily assigned in increasing order of the tail numbers.

A fundamental feature of the TAP is the satisfaction of maintenance requirements. To achieve this it is vital to identify the locations of maintenance critical aircraft at the start of each day. In the performed experiments the maintenance plan is randomly generated: Assigning each aircraft to receive maintenance exactly once in a six-day period. An extensive review is performed by solving the TAP using 100 randomly generated maintenance plans.

The objective of the TAP is to minimise the number of maintenance misalignments and costs associated with any gate reassignments. The slack variables; s_1^r, $b \in B, r \in R^b$, s_2^b and $s_3^b, b \in B$; are used to identify the maintenance misalignments on days one, two and three respectively. The maintenance misalignments are penalised in objective (24) using the cost parameters $c_1^r = 10\,000\ \forall b \in B, \forall r \in R^b$, $c_2^b = 7000\ \forall b \in B$ and $c_3^b = 4000\ \forall b \in B$. The second objective of the TAP—minimising the gate reassignment costs—is related to the real actions performed at an airport during daily operations. The largest cost of gate reassignment is the towing of aircraft between gates, which is commonly charged at a fixed rate. This is modelled in objective (24) by the parameters $c'_{ij} = 5000\ \forall b \in B, \forall i \in G^b, \forall j \in G^b$, which is an estimate of the real cost. Changing the gate assignment for set LOFs typically does not incur a cost during the day of operations. Hence, the parameters $c_{ij} = 1000\ \forall b \in B, \forall i \in G^b, \forall j \in G^b$ in objective (24) are a measure of inconvenience for the operations controllers regarding all changes related to ancillary services following a gate reassignment.

5 Computational experiments

The computational experiments performed assess the ability to satisfy maintenance requirements by solving the TAP using fixed LOFs and the improvements achieved by employing the iterative algorithm. Section 5.1 presents the former set of experiments using different LOF inputs and maintenance schedules. The results show a prevalence of maintenance misalignments, justifying the development of the iterative algorithm presented in Section 3.

The performance of the iterative algorithm—in regards to run time and solution quality—is presented in Section 5.2. Comparisons and evaluations of the maintenance misalignments, run time and objective values are made between the standard TAP formulation, iterative algorithm and a full column generation algorithm. The column generation algorithm relies on the complete model of the TAP, i.e. (24), (2)–(23), and generates variables for all aircraft in R using the column generation subproblem given by (26)–(30) with $\hat{N} = N$ and $\hat{C} = C$.

The experiments are performed using the SCIP Optimisation Suite 3.2.0, which includes SCIP 3.2.0 and SoPlex 2.2.0 [1]. The computing infrastructure used for the experiments consists of a cluster of Intel Xeon X5672 CPUs with 3.20 GHz and 48 GB RAM, running Ubuntu 14.04. Each experiment was performed on a single thread exclusively on one node.

5.1 Analysing the improved maintenance planning

Experiments are conducted on the TAP to assess the number of maintenance misalignments when using the two different LOF inputs described in Section 4. The 100 randomly generated maintenance plans are used to examine the robustness of the TAP to different operating conditions. In regards to the input LOFs, the results of Maher et al. [13] suggest that the use of the aircraft routing LOFs should cause maintenance misalignments in the TAP solution. In contrast, the SDAMRP LOFs are constructed by [13] to significantly reduce—or completely eliminate—the maintenance misalignments. The experiments presented in this section will assess the impact of the LOF and maintenance plan input.

The results presented in Figure 3 demonstrate a decrease in the number of maintenance misalignments as a result of using the SDAMRP LOFs compared to the aircraft routing LOFs. While the SDAMRP LOFs are shown by [13] to reduce the maintenance misalignments on average, in practice many misalignments still exist. This is shown by the light grey bars in Figure 3 in the columns representing at least one misaligned aircraft. As such, it is clear that simply using SDAMRP LOFs does not alone result in the complete elimination of day one maintenance misalignments. As a result some over-the-day swaps are necessary for many realisations of the maintenance plan.

A summary of the number of maintenance misalignments on days one, two and three after solving the TAP is presented in Figure 4. These results aim to demonstrate the influence of key features of the TAP—the day-

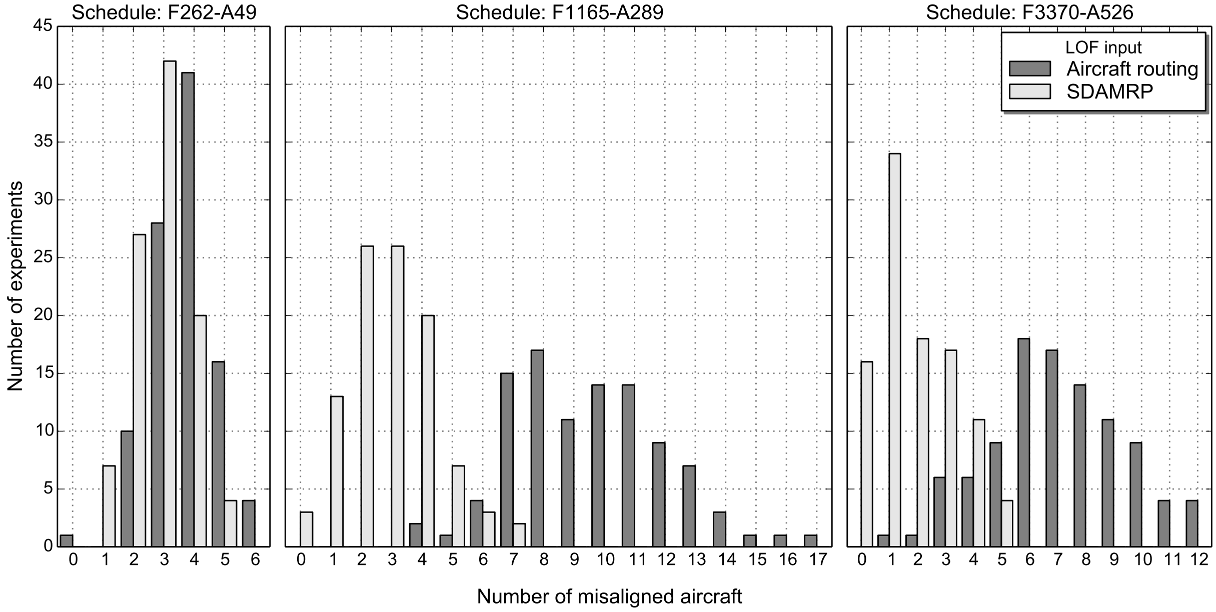
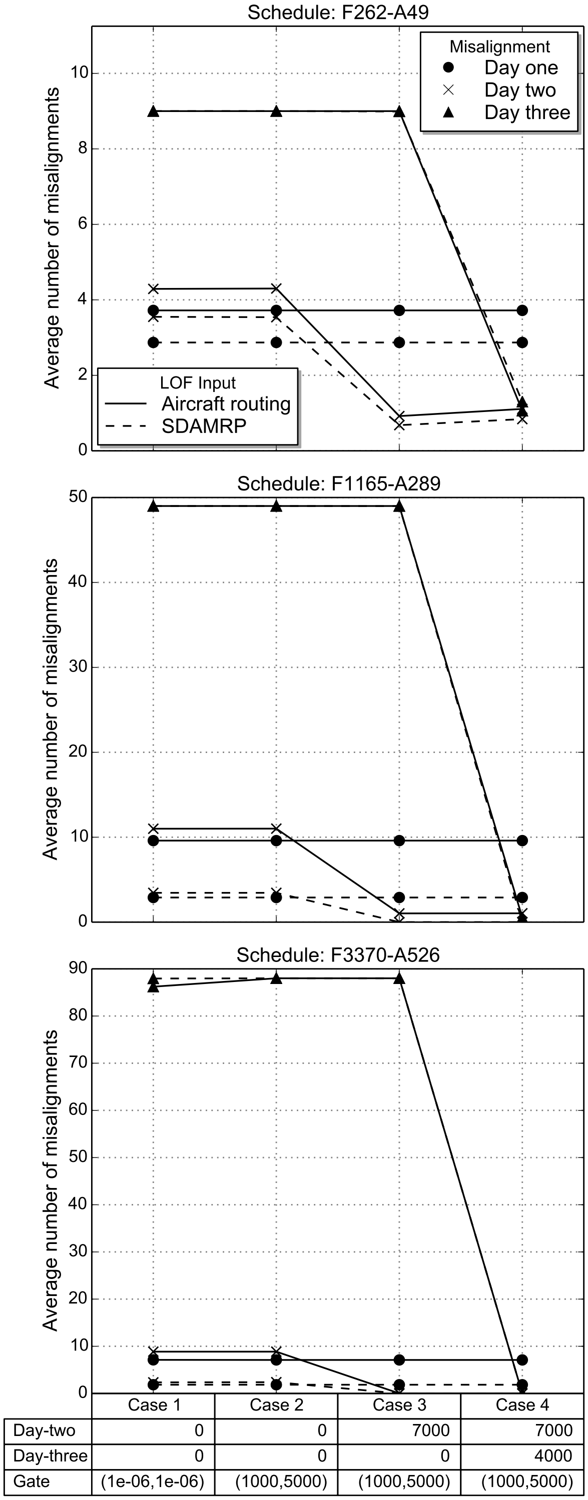


Figure 3: Histogram presenting the number of maintenance misaligned aircraft at the end of day-one over a set of 100 experiments solving the TAP with fixed LOFs.



	Case 1	Case 2	Case 3	Case 4
Day-two	0	0	7000	7000
Day-three	0	0	0	4000
Gate	(1e-06,1e-06)	(1000,5000)	(1000,5000)	(1000,5000)

Figure 4: Average maintenance misalignments on days one, two and three using various penalty settings to solve the TAP. Gate penalties (x, y) indicate the LOF and aircraft movement respectively.

two and day-three look-ahead maintenance constraints and the gate assignment constraints—on maintenance misalignments. The first column of Figure 4 represents the solution to a model when considering only day-one maintenance misalignments. The second considers the impact of introducing gate assignment penalties. The third presents the impact of penalising day-two maintenance misalignments. Finally, the fourth presents the results from solving the TAP by additionally penalising day-three maintenance misalignments.

A striking observation from Figure 4 is the little interaction between the different features of the TAP. The increase in the penalty values only impacts the feature directly affected. For example, it is only possible to reduce the day-two or day-three maintenance misalignments through the direct consideration with look-ahead constraints. As such, solving the TAP to only minimise day-one maintenance misalignments will result in infeasibilities of the maintenance plan on subsequent days. This result suggests that the developed TAP is valuable for reducing the number of day-one, day-two and day-three maintenancemisalignments.

The selection of LOF input is observed to be critical across all metrics presented in Figure 4. The SDAMRP LOFs outperform the aircraft routing LOFs in all metrics except the day-three maintenance misalignments in Case 1 for the F3370-A526 schedule and Case 4 for the F267-A49 schedule. Importantly, the number of day-one maintenance misalignments is significantly reduced using the SDAMRP LOFs compared to the aircraft routing LOFs. Solving the TAP with the SDAMRP LOFs achieves a reduction in the average number of day-one maintenance misalignments of 22.85%, 69.79% and 73.94% for the F267-A49, F1165-A289 and F3370-A526 flight schedules respectively.

Figure 4 presents a large decrease in the number of day-two and day-three maintenance misalignments when solving the TAP using the standard settings (Case 4). The average number of day-three maintenance misalignments using the aircraft routing and SDAMRP LOF input decreases from (9, 9), (49, 49), (88, 87.95) in Case 1 to (1.06, 1.3), (0.53, 0) and (0, 0) in Case 4 for the F267-A49, F1165-A289 and F3370-A526 flight schedules respectively. Similar results are observed for the day-two maintenance misalignments. However, the decrease is not as significant. These results suggest the need to consider subsequent days maintenance requirements when using one-day routes as input for the tail assignment.

An interesting observation from the solution of the TAP is that there are cases where the day-three maintenance requirements can not be satisfied. This situation only occurs for experiments using the aircraft routing LOF input. In some experiments there does not exist a feasible number of LOFs arriving at a maintenance station on day three. As a result, it is impossible to satisfy the day-three maintenance requirements using the provided LOFs. Since the day-three requirements are not met, this implies adjustments to the LOFs are required. This is primarily achieved using many over-the-day swaps, which is a very costly option for the airline.

5.2 Evaluating the iterative algorithm

The TAP and solution methods developed in this paper attempt to reduce the number of maintenance misalignments and gate assignment changes. The results presented in Section 5.1 show the inability of the TAP to completely eliminate maintenance misalignments using fixed LOF input. The route adjustment procedure developed in Section 3 is designed to address this limitation of the TAP.

The route adjustment procedure—implemented as an iterative algorithm—is expected to reduce the maintenance misalignments using an appropriately small amount of computational effort. The performance of the iterative algorithm is assessed with comparisons to the TAP and a column generation algorithm. A comparison of the day-one maintenance misalignments resulting from the use of each algorithm is presented in Section 5.2.1. The computational performance of the solution algorithms is assessed in regards to the solution run time. The run time comparison is presented in Section 5.2.2. Finally, the objective function of the TAP is composed of penalties related to maintenance misalignments and gate assignment changes, which are not evaluated in the assessment of day-one maintenance misalignments. The performance of each solution algorithm in regards to objective function value is presented in Section 5.2.3.

5.2.1 Maintenance misalignments

The comparison of the iterative algorithm and the TAP in regards to the day-one maintenance misalignments is presented in Figures 5 and 6. The histograms presented in Figures 5 and 6 were produced using the aircraft routing and SDAMRP LOFs respectively. Both Figures 5 and 6 show a significant decrease in the number of day-one maintenance misalignments by using the iterative algorithm. This effect is particularly evident for the F1165-A289 flight schedule using the aircraft routing LOF input. Solving the TAP using F1165-A289 flight schedule results in four to seventeen day-one maintenance misalignments. These misalignments are completely eliminated in all except five experiments, which only exhibit a single misalignment, when the iterative algorithm is employed. While this result is also observed when using the SDAMRP LOF input, the decrease in day-one maintenance misalignments is not as great. This is a consequence of the TAP solution using SDAMRP LOF input exhibiting less day-one maintenance misalignments compared to the aircraft routing LOF input for all flight schedules.

The success of the iterative algorithm is evident in the maximum number of day-one maintenance misalignments across the 100 experiments. For the aircraft routing LOF input the iterative algorithm achieves a maximum of 2, 1 and 5 maintenance misalignments for the F267-A49, F1165-A289 and F3370-A526 flight schedules respectively. Similarly, for the SDAMRP LOF input the maximum number of maintenance misalignments is 2, 1 and 2 respectively. While the maintenance misalignments are not completely eliminated in all experiments, the decrease is practically significant. Performing one or two over-the-day swaps is much

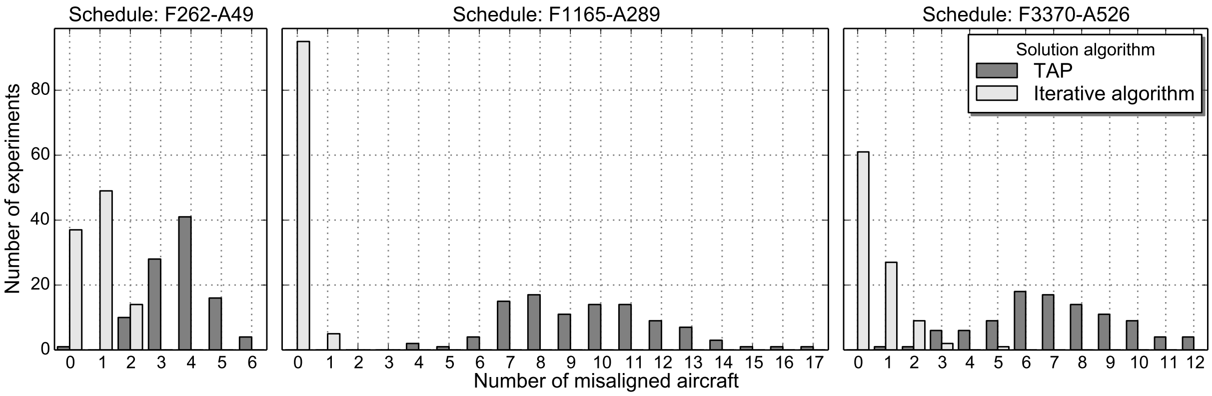


Figure 5: Comparing the TAP and iterative algorithm with histograms of the number of maintenance misaligned aircraft at the end of day-one over a set of 100 experiments using the aircraft routing LOF.

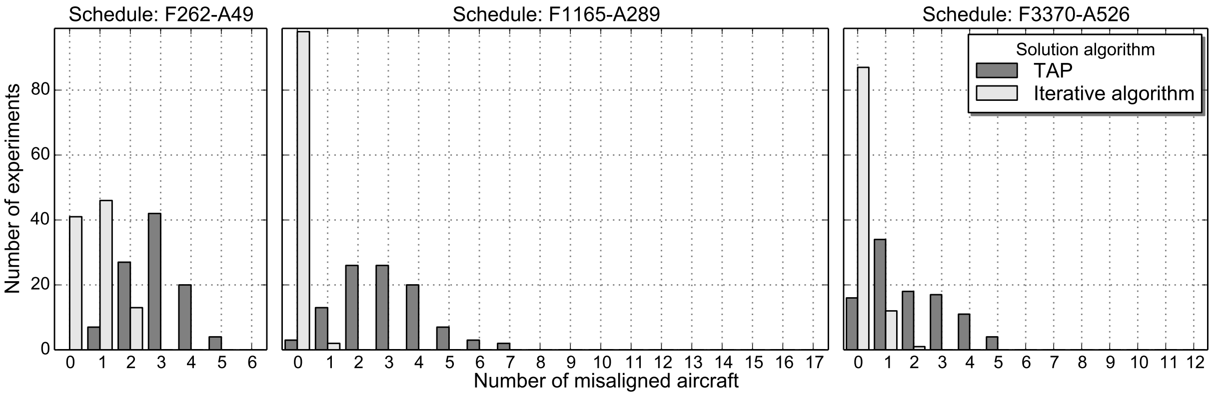


Figure 6: Comparing the TAP and iterative algorithm with histograms of the number of maintenance misaligned aircraft at the end of day-one over a set of 100 experiments using the SDAMRP LOF.

simpler for the airline than rerouting the large number of misaligned maintenance critical aircraft given by the TAP solution.

A surprising result is observed with the F3370-A526 flight schedule. The iterative algorithm using the aircraft routing LOFs still results in five maintenance misalignments. This number of maintenance misalignments may require prohibitively many actions by operations control centre to satisfy maintenance requirements. In comparison to the results for the iterative algorithm using the SDAMRP LOF input, the F3370-A526 flight schedule exhibits at most two maintenance misalignments. This demonstrates the importance of the LOF generation in aiding the satisfaction of maintenance requirements for both the TAP and iterative algorithm.

Similar to the preceding discussion Figures 7 and 8 present a comparison of the day-one maintenance misalignments when using the iterative algorithm and a column generation algorithm. The first observation from Figures 7 and 8 is that employing column generation for the F267-A49 flight schedule results in less maintenance misalignments compared to when the iterative algorithm is used. However, it must be noted that the column generation algorithm does not dominate the iterative algorithm across the set of 100 experiments. While the column generation algorithm achieves a better average day-one maintenance misalignment result, the small number of misalignments achieved by both algorithms is acceptable for the tail assignment application.

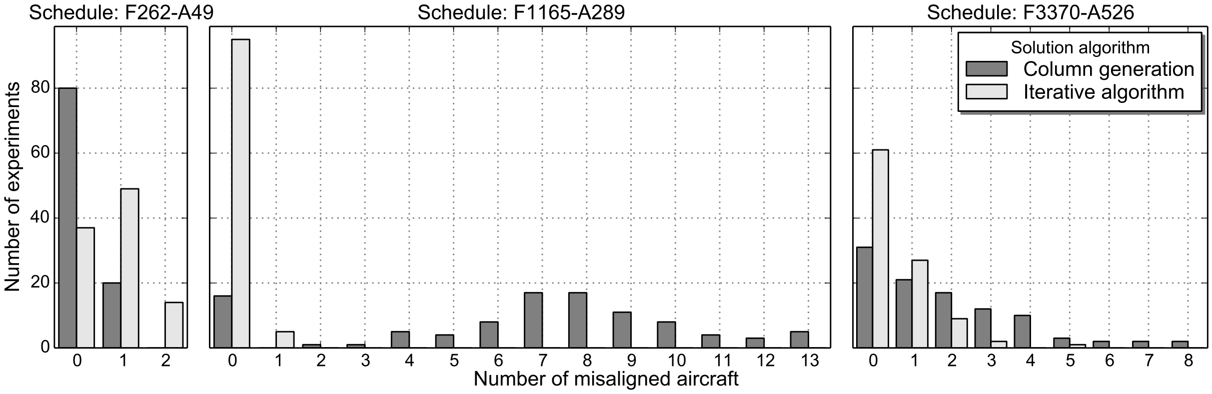


Figure 7: Comparing the column generation and iterative algorithms with histograms of the number of maintenance misaligned aircraft at the end of day-one over a set of 100 experiments using the aircraft routing LOF.

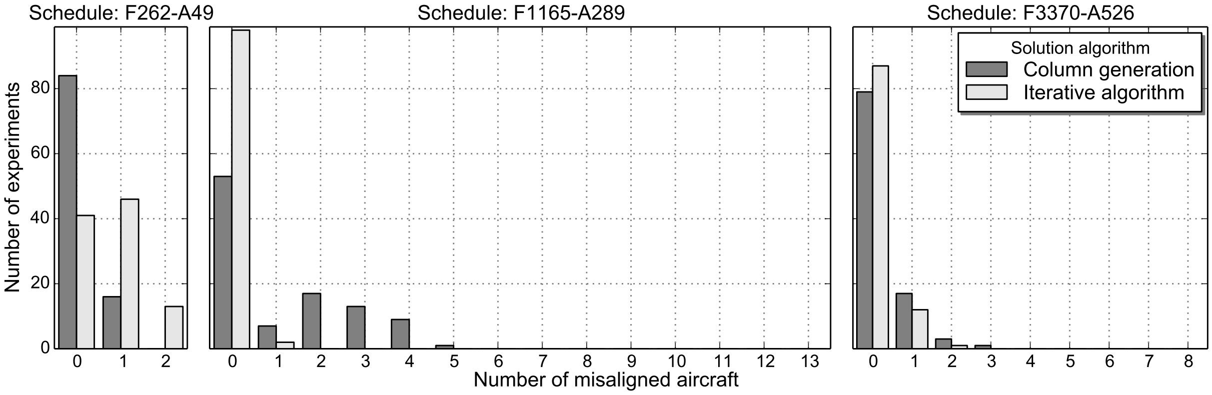


Figure 8: Comparing the column generation and iterative algorithms with histograms of the number of maintenance misaligned aircraft at the end of day-one over a set of 100 experiments using the SDAMRP LOF.

Contrary to the results for the F267-A49 flight schedule, the iterative algorithm outperforms the column generation algorithm for the F1165-A289 and F3370-A526 flight schedules in regards to day-one maintenance misalignments. This can be explained by the column generation algorithm failing to find the optimal solution within the maximum run time of 7200 seconds (2 hours) for a large proportion of experiments. Specifically, the column generation algorithm exceeds the time limit for 85 and 51 experiments using the aircraft routing and SDAMRP LOFs for the F1165-A289 flight schedule respectively and similarly 89 and 71 experiments using the F3370-A526 flight schedule. In such cases the best found feasible solution is reported. The reported solutions are of high quality compared to those achieved by the TAP, but are weaker on average compared to the solution of the iterative algorithm. The results presented in Figures 7 and 8 and the superior computational performance of the iterative algorithm demonstrates the value of employing such a heuristic approach to solve the TAP.

5.2.2 Solution approach run time

The run times for the 100 experiments with the TAP, iterative and column generation algorithms using the aircraft routing and SDAMRP LOF inputs are presented in Figure 9. A maximum run time of 7200 seconds is used for all experiments, which is deemed an appropriate limit for the considered application. All experiments for the TAP and iterative algorithm terminate within the maximum run time, while the column generation algorithm exceeds the maximum run time for many experiments.

An important observation from Figure 9 is the very short run times for the TAP algorithm. All experiments for the TAP algorithm using the F267-A49, F1165-A289 and F3370-A526 flight schedule terminate in, respectively, 0.8, 15.5 and 28.2 seconds for the aircraft routing LOF input and 0.5, 15.5 and 28.5 seconds for the SDAMRP LOF input. Comparatively, the iterative algorithm requires much longer run times. The iterative algorithm requires at most 6.8, 1103.5 and 1177.2 seconds for the F267-A49, F1165-A289 and F3370-A526 flight schedules respectively. While this represents a significant increase in run times, it is acceptable given the observed decrease in the number of maintenance misalignments achieved by the iterative algorithm as presented in Figures 5 and 6.

The benefit of the iterative algorithm is demonstrated by comparing the run times with that of the column generation algorithm. Figure 9 shows the iterative algorithm achieves a smaller run time on average compared

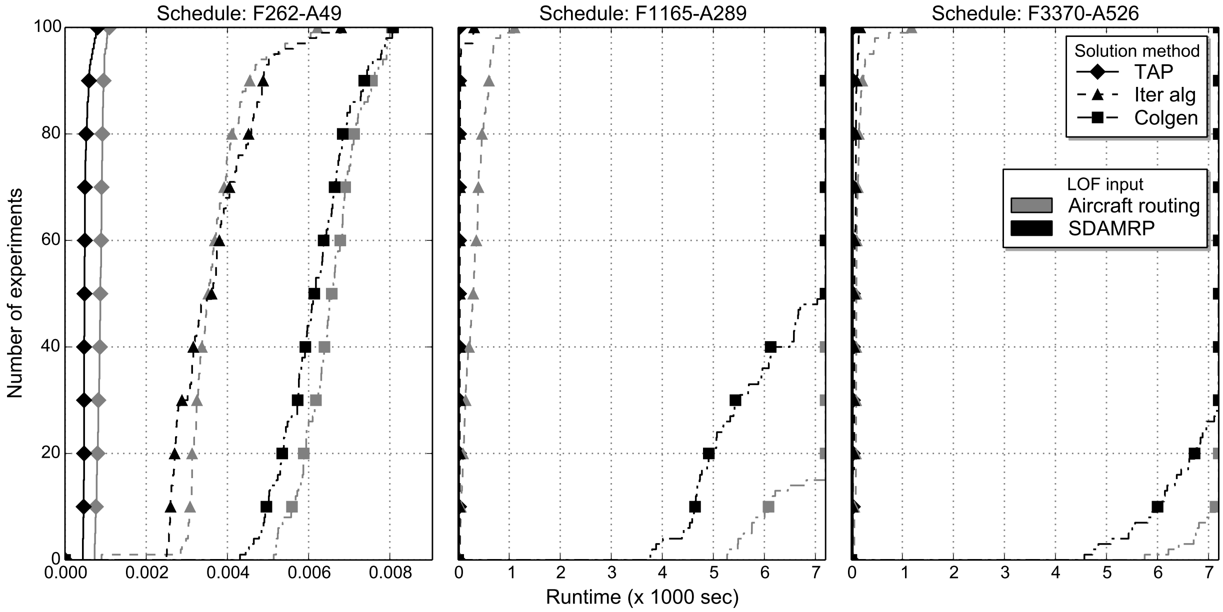


Figure 9: The run time required to solve 100 tail assignment instances using the TAP, iterative algorithm and column generation algorithm.

to column generation across all flight schedules. More importantly, the column generation algorithm fails to find the optimal solution within the maximum run time for most experiments using the F1165-A289 and F3370-A526 flight schedules. The maintenance misalignment reduction shown in Figures 7 and 8 and run time comparison given by Figure 9 demonstrates that the iterative algorithm is more practically useful for reducing the number of day-one maintenance misalignments compared to the column generation algorithm.

5.2.3 Optimal objective value

The analysis presented in Section 5.2.1 comparing the solution algorithms focuses only on the day-one maintenance misalignments. However, the objective function (24) is additionally composed of costs related to the day-two and day-three maintenance misalignments and gate assignment changes. As such, an improvement in the day-one maintenance misalignments may not result in an improved objective function value. Figure 10 presents the objective function values for all experiments with each solution algorithm using the aircraft routing and SDAMRP LOF inputs.

It is observed in Figure 10 that the iterative algorithm achieves a significantly lower objective function value compared to the TAP. This is an expected result given the magnitude of the maintenance misalignment reduction presented in Figures 5 and 6. While the decrease in day-one maintenance misalignments greatly contributes to the objective value improvement, there are mixed effects observed for the other cost components. In particular, employing the iterative algorithm results in an increase in the number of gate assignment changes on average across all experiments. This result can be explained by the route adjustment problem being formulated without the gate assignment constraints. Hence, the increase in the gate assignment changes is a trade off for the valuable reduction in the day-one maintenance misalignments.

The comparison between the iterative algorithm and the column generation algorithm presents many interesting results. First, the F267-A49 flight schedule demonstrates a limitation of the iterative algorithm. Specifically, the column generation algorithm achieves a better objective value, indicating that the iterative algorithm does not solve the TAP to optimality. While optimality is not achieved, the solution using the iterative algorithm presents an important reduction in the day-one maintenance misalignments as demonstrated in Section 5.2.1.

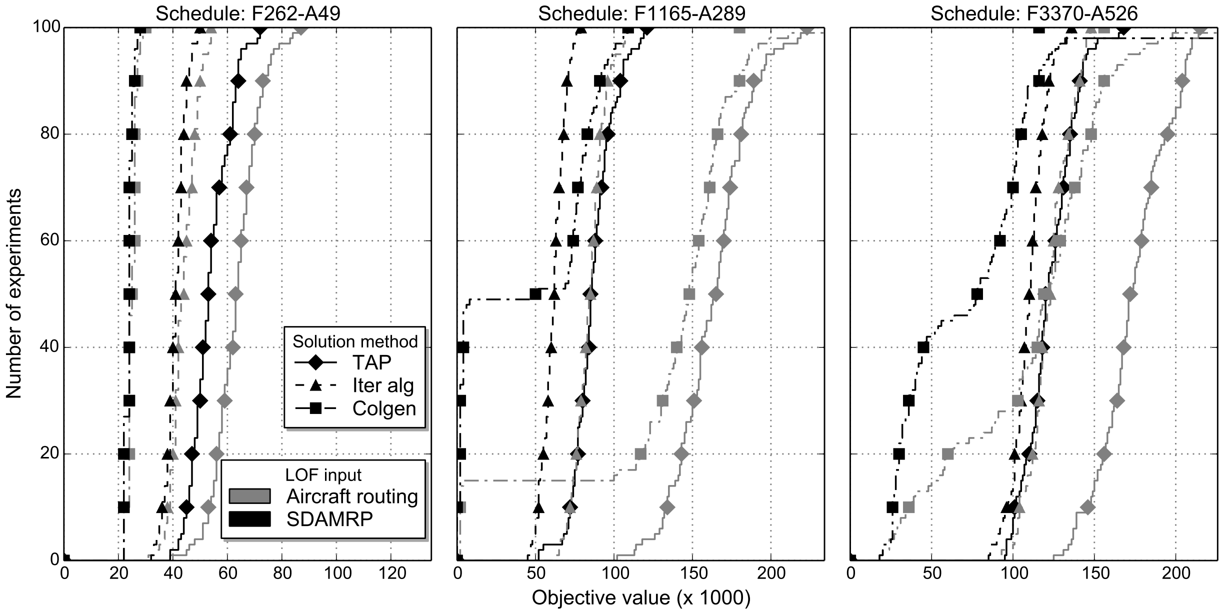


Figure 10: The objective function value profile over 100 tail assignment instances using the TAP, iterative algorithm and column generation algorithm.

A similar result is observed for the F1165-A289 flight schedule: The column generation algorithm achieves the lowest objective value for the experiments that terminate within the maximum run time. However, for all but one experiment where column generation exceeds the time limit the iterative algorithm demonstrates a better performance. Also, the column generation algorithm is unable to achieve much improvements over the TAP algorithm in the latter experiments.

The experiments using the F3370-A526 flight schedule presents varied results when comparing the iterative algorithm and column generation algorithm. Using the SDAMRP LOF input the column generation algorithm achieves a better objective value than the iterative algorithm for 89 experiments. However, the iterative algorithm is observed in Figures 7 and 8 to be very competitive in regards to reducing the day-one maintenance misalignments. This result can also be explained by the formulation of the route adjustment problem without the gate assignment constraints. Additionally, using the aircraft routing LOF inputs with the F3370-A526 flight schedule there is no clear dominating algorithm in regards to the objective function value. Column generation achieves a better result than the iterative algorithm in 52 experiments. However, the results in Section 5.2.2 demonstrate the strength of the iterative algorithm to achieve high quality results in relatively short run times.

6 Conclusions

The TAP is an important and necessary stage in the airline planning process. Two fundamental considerations of the TAP are directly reviewed in this paper—maintenance planning and over-the-night gate assignments. The TAP is developed to minimise the number of maintenance misalignments on days one, two and three. Additionally, any changes to the line-of-flight gate assignments or aircraft parking locations is minimised. To further reduce the number of maintenance misalignments, an iterative algorithm is presented. This algorithm employs a column generation algorithm that only generates aircraft routes for a subset of the considered fleet. The results demonstrate that the TAP and iterative algorithm reduce the number of maintenance misalignments for a given set of LOFs.

The TAP is formulated with constraints that implicitly satisfy maintenance requirements on days two and three. This implicit method is shown to be an effective method to plan maintenance with a model that is solvable by state-of-the-art mixed integer programming solvers. Further, this modelling approach can be generalised to maintenance planning problems that span across multiple time periods. The maintenance misalignments resulting from solving the TAP indicate the need to develop a route adjustment process. The iterative algorithm is shown to require less computational effort than a column generation algorithm. The strength of the iterative algorithm is the use of mixed-integer programming solvers to guide the search for improving aircraft routes.

The integration of tasks within the airline planning process is a critical development for high quality solution approaches. The TAP is an example of one type of integration that combines the tail assignment, look-ahead maintenance planning and over-the-night gate assignment problems. Future research involves identifying tasks for further integration in the TAP. Such tasks include achieving equal utilisation of the fleet and managing the non-uniform costs of aircraft of various ages. In addition, the iterative algorithm is demonstrated to effectively reduce the number of maintenance misalignments in comparison to the TAP and a full column generation approach. While the iterative algorithm achieves good feasible solutions in short run times, a gap to the optimal solution still exists. Future research is targeted at reducing this gap and improving the effectiveness of the iterative algorithm.

References

[1] T. Achterberg. SCIP: solving constraint integer programs. Mathematical Programming Computation, 1(1):1–41, 2009.

[2] C. Barnhart, N. L. Boland, L.W. Clarke, E. L. Johnson, G.L. Nemhauser, and R.G. Shenoi. Flight string models for aircraft fleeting and routing. Transportation Science, 32(3):208–220, 1998.

[3] R. Borndörfer, I. Dovica, I. Nowak, and T. Schickinger. Robust tail assignment. In Proceedings of the fiftieth annual symposium of AGIFORS, 2010.

[4] L. Clarke, E. Johnson, G. Nemhauser, and Z. Zhu. The aircraft rotation problem. Annals of Operations Research, 69:33–46, 1997.

[5] M. Dunbar, G. Froyland, and C.-L. Wu. Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. Transportation Science, 46(2):204–216, 2012.

[6] T. Feo and J. Bard. Flight scheduling and maintenance base planning. Management Science, 35(12):1415–1432, 1989.

[7] R. Gopalan and K. T. Talluri. The aircraft maintenance routing problem. Operations Research, 46(2):260–271, 1998.

[8] M. Grönkvist. The Tail Assignment Problem. PhD thesis, Chalmers University of Technology and Göteborg University, August 2005.

[9] A. Heinhold. Maintenance reachability at Southwest Airlines. In Proceedings of the AGIFORS Scheduling and Strategic Planning Study Group Meeting, Dallas, 2008.

[10] IBM CPLEX Optimizer. URL: `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`. Date Accessed: 04-02-2015.

[11] E. Lacasse-Guay, G. Desaulniers, and F. Soumis. Aircraft routing under different business processes. Journal of Air Transport Management, 16(5):258–263, 2010.

[12] M. Lapp and A. Cohn. Modifying lines-of-flight in the planning process for improved maintenance robustness. Computers & Operations Research, 39(9):2051–2062, 2012.

[13] S.J. Maher, G. Desaulniers, and F. Soumis. Recoverable robust single day aircraft maintenance routing problem. Computers & Operations Research, 51:130–145, 2014.

[14] S. Ruther. Integrated aircraft routing, crew pairing, and tail assignment. PhD thesis, The University of Newcastle, March 2013.

[15] Southwest Airlines. Airport information. Date Accessed: 1st September 2014.

[16] C. Sriram and A. Haghani. An optimization model for aircraft maintenance scheduling and re-assignment. Transportation Research Part A: Policy and Practice, 37(1):29–48, 2003.

[17] US Airways. Airports. Date Accessed: 1st September 2014.

[18] O. Weide, D. Ryan, and M. Ehrgott. An iterative approach to robust and integrated aircraft routing and crew scheduling. Computers & Operations Research, 37(5):833–844, 2010.