European Journal of Operational Research 264 (2018) 548–557

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

A compact optimization model for the tail assignment problem

Oumaima Khaled[a,d], Michel Minoux[b], Vincent Mousseau[a,*], Stéphane Michel[c], Xavier Ceugniet[c]

[a] Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Grande Voie des Vignes, 92290, Chatenay-Malabry, France
[b] Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, 4 Place Jussieu, 75005, Paris, France
[c] France Lab, IBM, 1681 Route Des Dolines, Les Taissounieres Hb2, 06560, Valbonne, France
[d] France Lab, IBM, Gentilly, France

ARTICLE INFO

Article history:
Received 28 May 2016
Accepted 20 June 2017
Available online 23 June 2017

Keywords:
OR in airlines
Tail assignment
Aircraft routing
Maintenance routing
Integer linear program

ABSTRACT

This paper investigates a new model for the so-called Tail Assignment Problem, which consists in assigning a well-identified airplane to each flight leg of a given flight schedule, in order to minimize total cost (cost of operating the flights and possible maintenance costs) while complying with a number of operational constraints. The mathematical programming formulation proposed is compact (i.e., involves a number of $0-1$ decision variables and constraints polynomial in the problem size parameters) and is shown to be of significantly reduced dimension as compared with previously known compact models. Computational experiments on series of realistic problem instances (obtained by random sampling from real-world data set) are reported. It is shown that with the proposed model, current state-of-the art MIP solvers can efficiently solve to exact optimality large instances representing 30-day flight schedules with typically up to 40 airplanes and 1500 flight legs connecting as many as 21 airports. The model also includes the main existing types of maintenance constraints, and extensive computational experiments are reported on problem instances of size typical of practical applications.

1. Introduction

The airline sector is characterized by high operational costs, strong regulations, and a complex planning process. In today's competitive environment a permanent concern for companies is to utilize the available resources (crew personnel, gates, airport slots, aircraft...) in the most efficient way. This fact has led airline companies to constantly seek for improved efficiency of the planning process through the use of advanced optimization models and tools.

The airline planning process is commonly divided into several subproblems, which are usually addressed and solved sequentially:

1. Timetable construction based on traffic forecast between cities/airports;
2. Fleet Assignment, which consists in deciding which fleet (i.e., aircraft type) is going to operate each flight in the timetable;
3. Crew planning (or: rostering) which is to assign crew personnel to the various flights;
4. Tail Assignment, which is to allocate a specific airplane (identified by its tail number) to each flight leg;
5. Recovery planning, which aims at adjusting the schedule in case of unexpected events (airport closure, absence of crew, delays, etc.).

For more details on the above overall planning process, refer to survey papers such as Barnhart et al. (2003); Gopalan and Talluri (1998).

In this paper, we are interested in the Tail Assignment problem, also referred to as aircraft routing problem (Clarke, Johnson, Nemhauser, & Zhu, 1997; Kabbani & Patty, 1992). As mentioned above, Tail Assignment is the problem of deciding which individual airplane should be assigned to each flight of a given timetable in order to optimize some objective function (minimizing assignment costs and/or maintenance costs) under a number of structural or operational constraints.

The main constraints to consider in the Tail Assignment problem are those specifying whether it is possible to successively operate two given flights with the same airplane. This implies compliance with (a) a minimum time condition; (b) an equipment continuity condition. Condition (a) aims at ensuring that a minimum time is left for cleaning, changing crew and passengers, and preparing aircraft for the next flight. Condition (b) enforces the

* Corresponding author.
E-mail addresses: oumaima.khaled@centralesupelec.fr (O. Khaled), michel.minoux@lip6.fr (M. Minoux), vincent.mousseau@centralesupelec.fr (V. Mousseau), stephane.michel@fr.ibm.com (S. Michel), CEUGNIET@fr.ibm.com (X. Ceugniet).

http://dx.doi.org/10.1016/j.ejor.2017.06.045
0377-2217/

arrival airport of the first flight to be the same as the departure airport for the second flight. Besides this, we have to take into account a set of “initial conditions” which impose that each airplane should start from some given initial airport.

In addition to the above basic constraints, other types of restrictions may arise such as:

- constraints prohibiting some specified airplanes to land on some specified airports for a variety of reasons (noise, unavailable equipment, etc.);
- various types of maintenance constraints, the introduction of such constraints is postponed to Section 5 of the present paper where it will be shown how they can be incorporated into our model;
- other constraints, less frequently considered, can also be mentioned such as achieving balanced flight duration among the airplanes, or limiting the number of landings in some specified airports per airplane (see e.g., Jünger, Elf, & Kaibel, 2003).

As will become apparent from the sequel, most of the above-mentioned constraints can be taken into account in our tail assignment model.

The paper is organized as follows. The new compact 0–1 linear programming formulation for a basic version of the tail assignment problem (i.e., without handling of the maintenance constraints) is presented and discussed in Section 3. Section 4 is devoted to an extensive computational study of the model. Results are reported for exact solution of instances featuring up to about 1500 flight legs and a fleet with up to 40 airplanes. Section 5 then discusses the various types of maintenance constraints and shows how to include the most important ones into the model at the expense of only moderate increase in problem size. Extensive computational results on this version of the model are also reported.

2. Literature review and comparison with existing models and solutions techniques

In the last two decades, the Tail Assignment problem and various closely related fleeting problems have been extensively studied in the Operations Research literature, see for instance Klabjan (2005); Sherali et al. (2006).

There are three main mathematical programming formulations underlying most existing work, namely:

(a) the so-called set packing/partitioning model which is *non-compact* in the sense that it involves a number of decision variables depending exponentially on the problem size parameters. In these models, each 0–1 decision variable corresponds to a string, i.e., an ordered sequence of flights that originates and terminates at airports with maintenance facility. Since the number of feasible strings is usually much too large to be handled explicitly by problem solvers, it requires the use of the so-called *column generation technique* (additional strings being generated as needed in the course of the optimization process). If exact optimality is required, column generation has to be combined with Branch and Bound (this is referred to as ‘Branch and Price’ see e.g., Barnhart et al., 1998).

(b) the so-called integral multi-commodity flow formulation (Levin, 1971) which is a *compact model* in the sense that its size (number of 0–1 decision variables and number of constraints) is a polynomial function of the problem size parameters (number of aircraft, number of flight legs);

(c) the so-called time-space network model , first proposed in (Hane et al., 1995) to model the fleet assignment problem and subsequently generalized to aircraft routing and crew pairing problems (Clarke, Hane, Johnson, & Nemhauser, 1996). In the time-space network, each station is represented by a time line spanning the time horizon considered and each node in the time-space network denotes an event of departure or arrival of a flight at a station at a specific time; to allow connection between flights, a minimum *ground time* is added to the actual flight arrival time to obtain the time of arrival associated with the corresponding arrival node. Connections between the nodes involve three types of arcs, namely: ground arcs, flight arcs and overnight arcs. This formulation also gives rise to a *compact model* with large number of variables and constraints. To reduce the size of the formulation, two preprocessing methods can be used, namely node aggregation and island isolation (see e.g., Hane et al., 1995; Sherali et al., 2006).

The *column-generation approach* related to (a) has been extensively used to solve airline operations problems. This approach was initiated by the work of Lavoie, Minoux, and Odier (1988); Minoux (1984) on the crew pairing problem and then followed by many studies in various airline fields specially in fleet assignment, aircraft routing and robust aircraft routing (Barnhart et al., 1998; Daskin & Panayotopoulos, 1989; Desaulniers, Desrosiers, Dumas, Solomon, & Soumis, 1997; Grönkvist, 2005; Hane et al., 1995; Kabbani & Patty, 1992; Sarac et al., 2006). Cordeau, Stojković, Soumis, and Desrosiers (2001); Mercier, Cordeau, and Soumis (2005) apply column generation technique starting from a multi-commodity flow model with non linear resource constraints representing maintenance requirements. More recently, the column generation approach has been proposed for addressing robust versions of the tail assignment problem in Borndörfer, Dovica, Nowak, and Schickinger (2010) and Froyland, Maher, and Wu (2013). All these formulations are *noncompact* and thus completely different from the one proposed in the present paper.

The second type of model (item (b)) is a compact formulation based on a *multi-commodity flow* problem first proposed by Levin (1971) in the context of aircraft routing and fleeting problems. The resulting integer linear program uses {0, 1} decision variables (x_{ij}^k) which take the value 1 if and only if flight i and j are successively operated by airplane k corresponding to a sequence between two successive flights i and j. The number of variables in this problem is $O(|F|^2|P|)$ where $|F|$ and $|P|$ are respectively the total number of flights in the timetable and the total number of airplanes.

As compared with Levin's model, our model features much reduced size. More precisely, carrying out the comparison on the 40 instances solved in Section 4, we find a size reduction factor ranging from 32 to 78, with an average value of 55. As a typical example, we mention the case of 10 instances composed of 1251 flight legs, 40 airplanes and 21 airports on average leading to 3,083,104 binary decision variables under Levin's model, while the new model only features 52,256 binary variables; the number of 0–1 variables in this example is reduced on average by a factor 59, thus enabling us to get exact optimal solutions within less than 1 hour CPU time.

The third widely used formulation (item (c)) is the so-called *time-space network model* which has been proposed by Hane et al. (1995) to handle fleet assignment problems. All the existing work based on time-space network models we are aware of address fleeting problems with an homogeneous fleet assumption together with additional constraints, which *we do not impose in our model*. Contrasting with this, in the present work the various airplanes considered are handled *individually*, which means that there is a cost specific to each of them, the initial position (airport) of each aircraft is taken into account; it is not required that the aircraft returns to its initial airport by the end of the time horizon considered (we refer to this as the ‘cyclic constraint’). This capability of handling each aircraft individually enables us for instance to

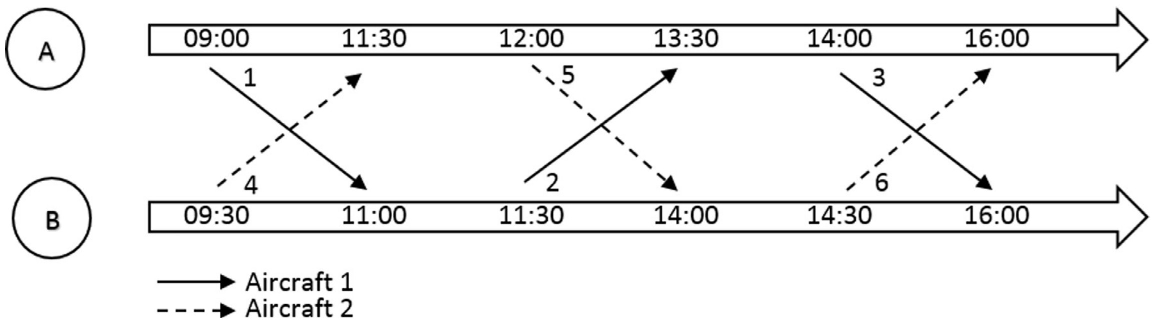

Fig. 1. Optimal flight schedule for example of Section 3.3.

properly address a situation frequently encountered by airline companies, namely the case of two aircraft which, in spite of being of the same type, differ by ages thus resulting in a significant difference w.r.t. fuel consumption (as is well-known, fuel consumption represents the essential part of the cost to be minimized).

Though part of the constraints in our model can be interpreted as combinations of flow-conservation constraints similar to those appearing in Hane et al. (1995) and other related works (Haouari, Shao, & Sherali, 2012; Liang & Chaovalitwongse, 2012; Liang, Chaovalitwongse, Huang, & Johnson, 2011), some distinctive features and advantages of our model are worth pointing out. Indeed, it is possible to re-express our model using the same set of decision variables as those involved in time-space network models (the variables expressing assignment of aircraft to flight sequences and the variables corresponding to ground arcs), but doing it results in a significant increase of the size of the model (both the number of variables and the number of constraints are typically augmented by a factor 1.25 to 2).

Another important distinctive feature is that, due to the fact that we do not impose all the constraints of a fleeting problem, in particular the requirement that the flight schedules determined to cover the various flight legs should lend themselves to cyclic repetition over time (the '*cyclic constraint*'), some more flexibility is left to find a feasible schedule. This fact is illustrated in Section 3.3 by means of a small example featuring 6 flight legs, 2 aircraft and 2 airports. For this example our model admits a feasible solution (Fig. 1), while imposing the cyclic constraint (as is done for instance in Hane et al., 1995) leads to an *infeasible problem*.

It is therefore clear from this analysis that, even if the models in Hane et al. (1995) and ours feature some structural similarity, they are different by size, by the solution space considered and by a number of constraints.

Finally we note that a close look at the most recent works related to the time-space network model leads to confirm the main distinctive features highlighted above. Liang et al. (2011) addressed a daily aircraft maintenance-routing problem by constructing a rotation-tour time-space network with wraparound maintenance arcs. Their model is of polynomial size but the size of the problem instances solved is limited to 352 flights. In a follow-on paper, Liang and Chaovalitwongse (2012) extended their previous model to consider the weekly aircraft maintenance routing problem by duplicating the daily network multiple times and linking them with proper maintenance arcs. Fleet assignment decisions are also included within this modeling framework by replicating the same routing network for each type of aircraft. However, the resulting large scale model is not solved exactly but only approximate solutions are obtained via variable fixing heuristics. In Haouari et al. (2012), the authors present a compact formulation for the aircraft routing problem including nonlinear constraints that are subsequently linearized by applying the reformulation–linearization technique (RLT). In addition to the fact that their model is quite different from ours (at least because it is nonlinear), the largest instances solved are limited to 344 flights only. All the above-mentioned works focus on daily or weekly repeated flight schedules therefore assuming the same "cyclic constraint" as in Hane et al. (1995).

In the present work, we introduce an improved compact formulation for the Tail Assignment problem featuring only $O(|F||P|)$ variables which, to the best of our knowledge, appears to be new. Our model, is of significantly reduced size as compared with other existing models, does not impose the cyclic constraint and handles the various airplanes individually. As will be shown through extensive computational experiments, the new formulation enables one to solve to exact optimality real size instances of the problem simultaneously taking into account all kinds of maintenance constraints with standard state-of-the art MIP solvers.

3. Modeling the Tail Assignment problem

3.1. Problem definition

The problem addressed is described by means of a timetable including a specified set of flight legs, where each fight leg i is defined by a departure airport $a_i^{\nearrow}$, an arrival airport $a_i^{\searrow}$, a departure time $t_i^{\nearrow}$ and an arrival time $t_i^{\searrow}$. n_f denotes the number of flight legs and the set of flight legs is $F = 1, \dots, i, \dots, n_f$. A typical timetable is shown in Table 2.

The Tail Assignment problem aims at defining for each specific airplane of a given fleet, the subset of flights which should be operated by this airplane in such a way that: (a) all flights of the timetable are covered; (b) total cost is minimized.

In this section, we limit ourselves to presenting first a basic version of the model which does not take maintenance constraints into account, the extended version including maintenance will be the subject of Section 5. Besides the fact that such a progressive approach is likely to make it easier to understand the model, we find it valuable to begin with highlighting what constitutes the generic part (the 'core') of the model, i.e., that part which is independent of a particular way of specifying the maintenance constraints.

The set of airplanes considered is indexed by $j = 1, \dots, n_p$, we set $P = \{1, 2, \dots n_p\}$. In the assignment process, each flight i from the timetable is assigned to an airplane $j \in P = \{1, \dots, j, \dots, n_p\}$ and the corresponding cost c_{ij} is supposed to be known and given. Even if, in practice, the tail assignment problem often concerns an homogeneous fleet (i.e., a set of airplanes having similar technical characteristics), the model to be proposed *doesn't rely on any homogeneity assumption*. In particular, if several distinct airplanes are capable of operating a given flight i, the associated costs are allowed to be different.

Table 1
Notation summary.

F	The set of flights: $F = \{1, \ldots, i, \ldots, n_f\}$
P	The set of airplanes to plan: $P = \{1, \ldots, j, \ldots, n_p\}$
A	The set of airports: $A = \{1, \ldots, k, \ldots, n_k\}$
$a_i^{\searrow}$	Arrival airport of flight i
$a_i^{\nearrow}$	Departure airport of flight i
$t_i^{\searrow}$	Arrival time of flight i
$t_i^{\nearrow}$	Departure time of flight i
Δt	Minimum time on ground needed between an arrival and a departure time
c_{ij}	Cost of assigning flight i to airplane j
a_j^0	The airport corresponding to the initial position of airplane j at the beginning of the horizon
$F_k^{\searrow}$	The set of flights which land in airport k
$F_k^{\nearrow}$	The set of flights which take off from airport k
$F_k^{\searrow}(\theta)$	The set of flight $i \in F_k^{\searrow}$ such that $t_i^{\searrow} \leqslant \theta - \Delta t$
$F_k^{\nearrow}(\theta)$	The set of flights $i \in F_k^{\nearrow}$ such that $t_i^{\nearrow} < \theta$

In the basic version of the Tail Assignment problem addressed in the present section (i.e., not considering maintenance), there are four main types of constraints that should be accounted for when assigning airplanes to flights:

(C1) *Flight coverage*: this classical constraint requires that each flight $i \in F$ should be covered by exactly one airplane $j \in P$.
(C2) *Equipment continuity*: airplane j can be assigned to flight i which departs from airport k only if it has previously landed in the same airport k early enough. This guarantees that airplane j is actually positioned in airport $k = a_i^{\nearrow}$ considering its previous flights.
(C3) *Initial conditions*: these constraints specify that every airplane j must depart from its initial position denoted a_j^0. In other words the departure airport of the first flight i operated by an airplane j must correspond to the initial position of the airplane: $a_i^{\nearrow} = a_j^0$.
(C4) *Turn time constraints*: the last type of constraints states that a minimum allowed time interval Δt, called either ground time or turn time, is necessary between the arrival and departure of an airplane. Thus, two flights i and i' can be carried out in sequence by the same airplane only if $a_i^{\searrow} = a_{i'}^{\nearrow}$ and $t_{i'}^{\searrow} \leq t_i^{\nearrow} - \Delta t$.

Definition 1. An assignment of airplanes to flights will be called a 'feasible flight plan' if it satisfies all the constraints 1–4 above.

3.2. Mathematical programming formulation

We now formulate the problem as a binary linear programming problem with $|F||P|$ 0-1 decision variables x_{ij}, where:

$$x_{ij} = \begin{cases} 1 & \text{if flight } i \text{ is operated by airplane } j \\ 0 & \text{else} \end{cases} \tag{1}$$

The objective is to minimize the total assignment cost $\sum_{i \in F} \sum_{j \in P} c_{ij} x_{ij}$ where, for each $i \in F$ and $j \in P$, the cost c_{ij} of assigning airplane j to flight i is known and given.

Using the notation subsumed in Table 1, the proposed model is stated as:

$$\textit{Minimize} \quad \sum_{i \in F} \sum_{j \in P} c_{ij} x_{ij} \tag{2}$$

s.t.

$$\sum_{j \in P} x_{ij} = 1, \quad \forall i \in F \tag{3}$$

$$\sum_{i' \in F_k^{\searrow}(t_i^{\nearrow})} x_{i'j} - \sum_{i' \in F_k^{\nearrow}(t_i^{\nearrow})} x_{i'j} \geq x_{ij} \qquad \forall j \in P,\ \forall k \in A \backslash \{a_j^0\}, \quad \forall i \in F_k^{\nearrow} \tag{4}$$

$$\sum_{i' \in F_k^{\searrow}(t_i^{\nearrow})} x_{i'j} - \sum_{i' \in F_k^{\nearrow}(t_i^{\nearrow})} x_{i'j} \geq x_{ij} - 1 \qquad \forall j \in P, \quad k = a_j^0, \quad \forall i \in F_k^{\nearrow} \tag{5}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in F, \quad j \in P \tag{6}$$

The objective function (2) represents the cost of assigning airplanes to flights. The *flight coverage* constraints (3) ensure that each flight i should be assigned to a single airplane j. Constraints (4) and (5) aim at ensuring both conditions (C2) (equipment continuity) and (C4) (turn time constraints) as explained below. Let $\underline{t} = \min_{i \in F} t_i^{\nearrow}$, and $\bar{t} = \max_{i \in F} t_i^{\searrow}$.

Given any solution x to (3)–(6), for any $\theta \in [\underline{t}, \bar{t}]$, for any $j \in P$ and $k \in A$, the expression:

$$B(j, k, \theta) = \sum_{i \in F_k^{\searrow}(\theta)} x_{ij} - \sum_{i \in F_k^{\nearrow}(\theta)} x_{ij}$$

represents the difference between the number of landings (not later than time $\theta - \Delta t$) and the number of take-offs (strictly before time θ) for airplane j and at airport k (net balance between landings and take-offs). With this notation, constraint (4) for $j \in P,\ k \in A \backslash \{a_j^0\},\ i \in F_k^{\nearrow}$ expresses the fact that setting $x_{ij} = 1$ is only possible if $B(j, k, t_i^{\nearrow}) \geq 1$. And constraint (5) for $j \in P,\ k = a_j^0,\ i \in F_k^{\nearrow}$ expresses the fact that setting $x_{ij} = 1$ is only possible if: $1 + B(j, k, t_i^{\nearrow}) \geq 1$

It is readily realized that any feasible plan complying with Definition 1 necessary satisfies constraints (4) and (5). The following proposition shows that, conversely, any feasible solution to program (3)–(6) corresponds to a feasible plan.

Proposition 1. *Any feasible solution to the system (3)–(6) is a feasible plan.*

Proof. Let x denotes any feasible solution to (3)–(6) and for any airplane $j \in P$, let $S_j = \{i_1, i_2, \ldots, i_p\}$ denote the subset of flights i such that $x_{ij} = 1$ in this solution, ordered according to non decreasing values of departure time i.e:

$$t_{i_1}^{\nearrow} \leq t_{i_2}^{\nearrow} \leq \cdots\cdots \leq t_{i_p}^{\nearrow}.$$

We are going to show that all the conditions (C2),(C3), and (C4) for airplane j are satisfied. First, suppose that (4) does not hold, i.e., that $a_{i_1}^{\nearrow} \neq a_j^0$. Then in this case: $B(j, a_{i_1}^{\nearrow}, t_{i_1}^{\nearrow}) = 0$ and since $x_{i_1 j} = 1$ this shows that constraints (5) corresponding to $i = i_1$ is violated, contradicting our assumption that x satisfied (3)–(6). Thus (C3) holds true.

Suppose now that conditions (C2) and/or (C4) for airplane j are not satisfied, and let (i, i') denote the first pair of consecutive flights in the ordered sequence S_j such that at least one of the conditions (C2) or (C4) does not hold true.

First assume that this is so because (C2) is not satisfied, i.e., that: $a_i^{\searrow} \neq a_{i'}^{\nearrow}$. Two cases have to be examined separately, depending on whether $a_{i'}^{\nearrow} \neq a_j^0$ (case 1), or $a_{i'}^{\nearrow} = a_j^0$ (case 2).

In case 1, it is easily checked that $B(j, a_{i'}^{\nearrow}, t_{i'}^{\nearrow}) = 0$. Since $x_{i'j} = 1$, this implies that constraint (4) corresponding to $i = i'$ is violated, contradicting the fact that x is a solution to (3)–(6).

In case 2, it is easily checked that: $B(j, a_{i'}^{\nearrow}, t_{i'}^{\nearrow}) = -1$. Since $x_{i'j} = 1$, this implies that the corresponding constraints (5) for $i = i'$ is violated, again leading to a contradiction.

In view of the above, (C2) necessary holds, which means that $a_i^{\searrow} = a_{i'}^{\nearrow}$ and if (i, i') is the first pair of consecutive flights for which at least one out of (C2) and (C4) does not hold true, this is necessarily because (C4) does not hold true.

So, assume that (i, i') is such that $a_i^{\searrow} = a_{i'}^{\nearrow} = k$, and, because (C4) does not hold: $t_{i'}^{\nearrow} < t_i^{\searrow} + \Delta t$.

Table 2
Example of a flight timetable.

Flight i	$a^{\nearrow}(i)$	$a^{\searrow}(i)$	$t^{\nearrow}(i)$	$t^{\searrow}(i)$
1	A	B	1.10.2016 09:00	1.10.2016 11:00
2	B	A	1.10.2016 11:30	1.10.2016 13:30
3	A	B	1.10.2016 14:00	1.10.2016 16:00
4	B	A	1.10.2016 09:30	1.10.2016 11:30
5	A	B	1.10.2016 12:00	1.10.2016 14:00
6	B	A	1.10.2016 14:30	1.10.2016 16:30

Table 3
Assignment costs c_{ij}.

Flight i	Airplane		
	1	2	3
1	6804	6870	6771
2	4536	4580	4514
3	7216	7286	7181
4	1133	1144	1128
5	6185	6245	6155
6	5773	5829	5745

Again two cases have to be examined, depending on whether $k \neq a_j^0$ (case 3) or $k = a_j^0$ (case 4).

In case 3, it is easily checked that: $B(j, k, t_{i'}^{\nearrow}) = 0$. Since $x_{i'j} = 1$, this implies that constraints (4) corresponding to $i = i'$ is violated, and a contradiction results.

In case 4, it is easy to check that: $B(j, k, t_{i'}^{\nearrow}) = -1$. Since $x_{i'j} = 1$, this implies violation of constraint (5) corresponding to $i = i'$, from which a contradiction is obtained.

From the above it is concluded that all the conditions (C2), (C3), and (C4) are satisfied for airplane j, and since this is true for any $j \in P$, the proof of Proposition 1 is concluded. □

The following proposition states the number of variables and constraints in model (2)–(6) and shows that it is a *compact model.*

Proposition 2. *Model (2)–(6) contains $n_f \times n_p$ binary decision variables. The number of constraints is bounded by $O(n_f \times n_p)$, where n_f and n_p represent the number of airplanes and flights.*

Proof. Clearly the model involves $n_f \times n_p$ decision variables (x_{ij}). The number of constraints (3) is equal to n_f. The number of constraints (4) is bounded by $O(n_f \times n_p \times n_a)$ but is actually less than this upper bound because we just consider flights which only depart from airport k. However for constraints (5), only flights departing from the initial position of airplanes are considered (($k \in \{a_j^0\}$), the number of such constraints is bounded by $O(n_f \times n_p)$.

In practice, the number of flights is more important than the number of airplanes which is also bigger than the number of airports so we can conclude that $n_f \gg n_p > n_a$ and the number of constraints in our mathematical formulation is $O(n_f \times n_p)$. □

As already pointed out in the introduction section, a distinctive feature of our model is that it involves significantly fewer decision variables and constraints as compared to other compact models existing in the literature.

3.3. Illustrative example

To illustrate, we present the following example extracted from a real timetable of an airline company. The data set includes 2 airplanes and 6 flights presented in Table 2 and Table 3 shows assignment costs. There are two airports (A and B), the turn time is set equal to 30 minutes, and the "initial conditions" are airport *A* for airplane *1* and airport *B* for airplane *2*. The optimal flight schedule is shown in Fig. 1 and the flight sequences corresponding to the two airplanes in the optimal solution for this example are shown in Table 4.

Table 4
Optimal solution.

Airplane	Sequence of flights
1	1 → 2 → 3
2	4 → 5 → 6

This example is also useful to illustrate one of the significant differences between the model of the present paper and those addressing fleeting problems, in which the so-called 'cyclic constraint' is present. If we impose that each airplane returns to its initial airport at the end of the time horizon, then it is easily checked that the problem has no feasible solution.

4. Computational Study

In this section, we report on computational experiments aimed at assessing the relevance and efficiency of our compact model on realistic instances of the Tail Assignment problem.

4.1. Data-generation procedure

In order to carry out our experimental analysis and to investigate the main parameters influencing computation time (*size* and *density*), a data generation procedure producing test instances with controlled size and density has been designed (the concept of *density* of a problem instance will be made more precise below). It makes use of an existing large scale real data set from a domestic low cost airline in Asia (composed of 1580 flight legs over 30 days with 37 airports and 50 airplanes involved).

The data-generation procedure takes as input parameters:

- a real parameter $\delta \in [0, 1]$ called density of the instance;
- the first and last day of the desired time period (between 1 and 30), defining the horizon h;
- the number of airplanes (np).

The procedure constructs a subset of flights forming the timetable of the generated instance as follows:

(a) First we solve the problem of determining np_{min}, the minimum number of airplanes necessary to cover all the flights in the set $\widetilde{F}$ of flights which have to be carried out in the specified horizon h. The initial positions of various airplanes are assumed to be the same as for the whole real data set. This problem can easily be formulated as the search for a minimum flow in a network describing the possible connections between flights, with lower and upper capacity bounds on the arcs. It can be very efficiently solved by using linear programming or via a specialized network flow algorithm. We denote $\bar{P}$ the subset of airplanes used in the resulting optimum solution, $np_{min} = |\bar{P}|$, and for each airplane $j \in \bar{P}$, we denote $L_j \subset \widetilde{F}$ the subset of flights operated by this airplane.

(b) After checking that $np \in \{1, \ldots, np_{min}\}$, which is the case for all the instances of the test set as soon as $np \leq 40$, we perform the following : a subset $\widetilde{P}$ of np district airplanes is randomly selected out of $\bar{P}$ to compose the fleet corresponding to the instances generated;

(c) A subset $\widehat{P}$ of $\delta \times np$ airplanes is randomly chosen out of the set $\widetilde{P}$, and the set of flights composing the desired instance is finally obtained as : $\bigcup_{j \in \widehat{P}} L_j$.

Thus, the *density* of an instance is the ratio between the minimum number of airplanes necessary to cover all the flights of the timetable and the number of airplanes in the available fleet. Accordingly, a 100% dense instance is an instance for which the number of available airplanes is exactly equal to the minimum number of airplanes necessary to ensure feasibility. When the density is lower, more flexility is left to find feasible solutions and the problem may be expected to be easier to solve.

Our experiments will confirm this guess: for higher density, increased computation times will be observed. To the best of our knowledge, this is the first time the influence of the density parameter on the practical difficulty of solving this class of problems is systematically investigated.

4.2. Computational results

In order to assess the performance of the proposed tail assignment model (3)–(6), we have implemented it with Python 2.7 and Cplex Studio 12.6.2 All computations have been performed on a workstation running Intel(R) Core(TM) i7-3740QM CPU @ 2.7 gigahertz. Tests are carried out on 400 instances generated randomly using our data-generation procedure. Each instance corresponds to a timetable of h days of flights for np airplanes. The various instances considered in our experiments have been obtained by varying the value of the three parameters:

- the density δ: 95%, 100%.
- the number of airplanes np: 10, 20, 30, 40;
- the time horizon h: 7, 15, 21, 30 days.

The timetables obtained differ in terms of the value of parameter h and number of flights f. Actually, for the same value of np and h, a timetable generated with a lower density δ necessarily features a smaller number of flights.

The results presented in Tables 5 and 6 have been obtained by disabling the Cplex cut generation option. The reason is that the generation of cuts does not add too much to the results obtained; indeed, we could observe that there is almost no difference whatsoever between the CPU times and number of nodes obtained by activating or by disabling this parameter. On the other hand, with have included in the tables a comparison of the results obtained *with and without* activation of the Cplex 'presolve' option. The interest of carrying out such a comparison will be explained below.

The computational results obtained concern two series of instances corresponding to two different values of the density parameter ($\delta = 0.95$ (Table 5) and $\delta = 1$ (Table 6)). The figures appearing on each line are average values for 10 distinct randomly generated instances for each choice of the following parameters: number of airplanes; horizon (in days); number of flight legs; number of variables and constraints of the resulting MIP model; number of nodes required by the tree search procedure (and, in parenthesis the maximum number of nodes over the 10 instances); the value of the residual gap (when equal to zero, this means that an exact optimal solution has been obtained for each of the 10 instances solved); value of the integrality gap upon starting the Branch-and-Bound procedure. The main comments suggested by these results are the following:

1. We observe that there is a significant gain in time obtained while activating the presolve option. The reduction is about 30–50% of CPU time. Moreover for a number of instances, small differences in objective values are observed, the best values being obtained without presolve in a majority of cases. However, the observed discrepancy does not seem to be significant as compared with the value of CPLEX default tolerance parameter (10^{-4}).

The next comments will concern the results *without presolve option*.

2. The values of the integrality gaps (displayed in the column "Gap at root") are extremely low: 0.25% on average, and always less than 0.5%. These figures confirm that the compact formulation proposed in the present paper is indeed very strong i.e., its linear relaxation very closely approximates the integer 0–1 problem to be solved. A further confirmation of the strength of the formulation is provided in Table 8 where the average values of the percentage of integral variables in the optimal solutions to the linear relaxation are shown. In all instances considered this percentage is more than 85%.
3. The number of nodes examined in the tree-search procedure for getting exact optimal solutions are also observed to be extremely low, with an average value less than 15 and a maximum value 55 over all the instances solved; these figures are quite remarkable for such large sized 0–1 linear problems (featuring up to about 59,000 binary variables and 60,000 constraints), and can be viewed as further evidence of the strength of the formulation.
4. Even with the pre-solve option of CPLEX switched off, the computation times are rather reduced, typically less than 10 minutes for a majority of the instances, and not exceeding 1 hour for a few of the largest instances.

Table 5
Computational results for problem density $\delta = 95\%$($|P|$ = number of airplanes, $|H|$ = horizon (number of days), $|F|$ = number of flights).

|P|	|H|	|F|	# Var	# Const	Without presolve # nodes (Max)	Residual gap (%)	Gap at root (%)	CPU (sec)	Objective value	With presolve CPU (sec)	Objective value
10	7	206	2057	3600	0(0)	0.00	0.02	2.62	1,672,661	0.51	1,672,662
	15	301	3006	4549	0(0)	0.00	0.04	11.54	2,444,206	1.01	2,444,206
	21	339	3386	4929	0(0)	0,00	0.05	21.13	2,757,809	1.32	2,757,815
	30	321	3213	4756	0(0)	0.00	0.10	39.71	2,609,676	1.15	2,609,680
20	7	362	7248	7610	0(0)	0.00	0.05	6.80	2,948,290	6.00	2,948,289
	15	592	11,834	12,426	2(9)	0.00	0.07	46.99	4,842,097	24.43	4,842,140
	21	620	12,402	13,022	4(16)	0.00	0.12	61.46	5,076,054	25.10	5,076,054
	30	697	13,944	14,641	4(15)	0.00	0.15	72.84	5,709,698	35.81	5,709,752
30	7	574	17,223	17,797	2(6)	0.00	0.10	76.82	4,698,590	34.04	4,698,608
	15	914	27,426	28,340	3(6)	0.00	0.15	347.62	7,511,488	167.38	7,511,489
	21	999	29,973	30,972	4(9)	0.00	0.19	534.93	8,202,223	216.99	8,202,224
	30	1052	31,566	32,618	2(5)	0.00	0.32	636.80	8,641,467	249.40	8,641,523
40	7	746	29,828	30,574	4(16)	0.00	0.21	271.47	6,121,668	125.23	6,121,669
	15	1167	46,688	47,855	9(53)	0.00	0.25	1681.50	9,613,017	797.85	9,613,018
	21	1273	50,920	52,193	7(37)	0.00	0.30	2337.68	10,495,557	952.51	10,495,561
	30	1378	55,112	56,490	1(5)	0.00	0.34	3871.14	11,326,359	1919.52	11,326,361

Table 6
Computational results for problem density $\delta = 100\%$ ($|P|$ = number of airplanes, $|H|$ = horizon (number of days), $|F|$ = number of flights).

|P|	|H|	|F|	# Var	# Const	Without presolve # Nodes (Max)	Residual gap (%)	Gap at root (%)	CPU (sec)	Objective value	With presolve CPU (sec)	Objective value
10	7	230	2041	1542	0(0)	0.00	0.01	5.49	1,868,915	0.80	1,874,699
	15	417	3932	3344	0(0)	0.00	0.04	13.15	3,405,043	3.32	3,405,043
	21	496	4750	4043	0(0)	0.00	0.13	33.47	4,051,775	5.35	4,051,775
	30	574	5533	4600	1(5)	0.00	0.15	56.01	4,711,709	8.87	4,711,835
20	7	446	8922	9368	1(3)	0.00	0.20	18.03	3,639,950	10.63	3,639,950
	15	734	14,672	15,406	5(16)	0.00	0.24	110.65	6,023,260	38.88	6,024,530
	21	881	17,626	18,507	8(55)	0.00	0.27	184.53	7,360,725	72.05	7,360,961
	30	999	19,980	20,979	15(25)	0.00	0.30	263.11	8,223,005	94.09	8,226,049
30	7	623	18,696	19,319	4(6)	0.00	0.24	101.43	5,111,686	32.79	5,111,686
	15	1021	30,633	31,654	6(9)	0.00	0.30	616.92	8,397,863	198.58	8,397,863
	21	1178	35,337	36,515	4(5)	0.00	0.36	910.63	9,715,431	302.18	9,715,431
	30	1251	37,533	38,784	6(9)	0.00	0.42	1101.36	10,326,645	575,05	10,326,645
40	7	786	31,440	32,226	4(10)	0.00	0.27	318.68	6,451,054	100.97	6,451,054
	15	1225	49,012	50,237	6(15)	0.00	0.38	2069.98	10,121,282	673.23	10,121,282
	21	1306	52,256	53,562	10(18)	0.00	0.45	2986.84	10,758,751	987.09	10,759,131
	30	1494	59,764	61,258	5(5)	0.09	0.47	4864.96	12,252,010	2901.69	12,253,167

Table 7
Average CPU times (seconds) as a function of problem size (number of variables) and density.

Number of variables	Density $\delta = 0{,}9$	$\delta = 0{,}95$	$\delta = 1$
[1000–6000]	12.69	18.75	27.03
[6000–12,000]	15.27	26.89	28.59
[12,000–18,000]	62.45	70.37	147.59
[18,000–24,000]	128.45	153.45	182.27
[24,000–60,000]	956.73	1087.56	1195.68

Table 8
Proportion of integral variables in the optimal solutions to the linear relaxations (averages).

Number of variables	Density $\delta = 0{,}9$	$\delta = 0{,}95$	$\delta = 1$
[1000–6000]	98.6	98.1	98.2
[6000–12,000]	93.5	95.2	95.1
[12,000–18,000]	89.7	90.6	91.2
[18,000–24,000]	89.6	87.4	86.9
[24,000–60,000]	89.1	89.9	87.5

5. The influence of the density parameter δ on the practical difficulty of solving the problem exactly is illustrated in Table 7: for various ranges for problem size (number of variables) the average CPU times are shown for three values of the density parameter (0.9, 0.95, 1). It is seen that the computational effort significantly increases with δ. In practice we observe that for most airline companies, if the number of aircraft in the fleet supposed to carry out a given timetable is correctly adjusted, the corresponding value of the density parameter should typically be in the range 90–100%. As a typical example, considering the real data set used for generating our test instances (1580 flight legs over 30 days with 37 airports) the number of aircraft in the fleet is 50, while np_{min}, the minimum number of aircraft necessary to cover all the flights, is equal to 47, thus resulting in a density value of 94%.

5. Including maintenance constraints in the model

We now turn to discuss an extension of the basic model aimed at taking into account maintenance constraints. They correspond to requiring that all airplanes should regularly undergo various types of maintenance activities. These activities are of major importance for airline companies which have to implement schedules complying with the many types of security requirements involving maintenance; this is why many existing works in the literature dealing with aircraft routing include maintenance constraints i.e., aim at simultaneously optimizing assignment of aircraft to flights and assignment of aircraft to maintenance stations (see e.g., Lacasse-Guay, Desaulniers, & Soumis, 2010). Let us first briefly recall the different types of maintenance activities imposed by the aviation authority (FAA).

The are four major types of checks denoted A, B, C and D. These vary in scope, duration, and frequency (Clarke et al., 1997). The first major check (denoted as Type A) actually mandated by the FAA occurs at every 65 flight-hours, or about once a week. Type A checks involve inspection of all major systems such as landing gear, engines and control surfaces. However, the second major check (designated as Type B) is performed every 300 to 600 flight-hours, and entails a thorough visual inspection plus lubrication of all moving parts such as horizontal stabilizers and ailerons. The very major checks referred to as Type C and D are done about once every one to four years, respectively, and require leaving the aircraft out of service for up to a month at a time. Because Type B, C and D checks are spaced at relatively large time intervals and because of the dynamic nature of the market, these need not be taken into account in maintenance scheduling. The main concern of the airlines when solving aircraft routing and fleeting problems is in meeting Type A check requirements through their self-imposed 4–5 days inspection and maintenance policy. As a general rule, inspections and repairs take place at night.

To the best of our knowledge, there are only a few existing works in the literature that succeeded in *simultaneously handling all maintenance constraints of type A*. One can mention the string model of Barnhart et al. (1998), in which flight strings satisfying all the maintenance constraints can efficiently be generated via the use of constrained shortest path algorithms. Another approach, more recently proposed in Haouari et al. (2012) is based on a nonlinear (quadratic) mixed-integer model to solve the daily aircraft maintenance routing problem; the linearized model (obtained through application of the so-called 'Reformulation-Linearization Technique') can only solve medium-size instances (the largest instances considered involve up to 344 flights). In this section, we show how it is possible to include all these type-A maintenance constraints into our model *while preserving linearity of the resulting formulation*. Extensive computational results are presented and discussed in Section 5.3.

Table 9
Notation used for expressing maintenance constraints.

MA	The set of maintenance airports: $MA \subset A$
D	The set of days of the planning: $D = \{1, \ldots, d, \ldots, n_d\}$
$d_i^{\nearrow}$	Departure day of flight i
mc_{ijd}	Cost of assigning a night maintenance activity in the arrival airport of flight i to airplane j on day d
$mcap_{md}$	Maintenance capacity of airport $m \in MA$ on day d
$F(d, i)$	The set of flights i' where $t_{i'}^{\nearrow} > t_i^{\searrow}$ and $d_i^{\nearrow} = d$
$dmax$	The maximum allowed number of days for each aircraft without maintenance (typically 4 or 5 days)
$Tmax$	The maximum allowed flight time for each aircraft without maintenance (typically between 50 and 100 hours)
$M_{dd'}$	The big constant representing the maximum possible cumulated flight time for an airplane between days d and d'
$F_{dd'}$	The set of flights which depart between day d and day d'

5.1. Problem definition

In the model described here for maintenance, night maintenance activities are considered and thus, for each airplane, a maintenance activity on day d can only take place in the arrival airport of the last flight operated by the airplane on that day. We define $D = \{1, \ldots, d, \ldots, n_d\}$ the set of days of the planning and $MA \subset A$ the set of maintenance airports where maintenance activities could take place. Each maintenance airport $m \in MA$ is characterized by a capacity $mcap_{m,d}$ i.e., the maximum number of airplanes which can be handled on each day $d \in D$.

In the extended model, we show that all types of maintenance constraints which have been considered in the existing literature can be taken into account and efficiently handled by our model: essentially these correspond to what is referred to as type A checks, which have to be carried out:

- at least once every *dmax* days (where *dmax* is a parameter, typically between 2 and 6), this can be referred to as a '*maximum spacing constraint*';
- at least after no more than *Tmax* hours of flight (*Tmax* is a parameter typically chosen between 40 and 100) or at least after no more than ν take-offs (ν is a parameter typically chosen between 10 and 20), in both cases this is the so-called '*cumulative constraint*'.

5.2. Mathematical programming formulation

Maintenance incurs additional costs mc_{ijd} related to the night maintenance activity in the arrival airport of flight i when assigned to airplane j on day d.

The binary decision variables used in the formulation are $z_{ijd} \in \{0, 1\}$ where:

$$z_{ijd} = \begin{cases} 1 & \text{if a } night \text{ maintenance takes place in the arrival airport of flight } i \text{ and it is assigned to airplane } j \text{ on day } d, \\ 0 & \text{else} \end{cases}$$

$$y_{jd} = \begin{cases} 1 & \text{if maintenance of aircraft } j \text{ takes place on day } d, \\ 0 & \text{else.} \end{cases}$$

Using the additional notation to the basic tail assignment problem displayed in Table 9, the extended model is as follows:

$$Min \ \sum_{i \in F} \sum_{j \in P} c_{ij} x_{ij} + \sum_{m \in MA} \sum_{i \in F^{\searrow}(m)} \sum_{j \in P} \sum_{d \in D} mc_{ijd} z_{ijd} \tag{7}$$

s.t.

$$\sum_{j \in P} x_{ij} = 1, \quad \forall i \in F$$

$$\sum_{i' \in F_k^{\searrow}(t_i^{\nearrow})} x_{i'j} - \sum_{i' \in F_k^{\nearrow}(t_i^{\nearrow})} x_{i'j} \ge x_{ij}$$
$$\forall j \in P, \ \forall k \in A \backslash \{a_j^0\}, \ \forall i \in F_k^{\nearrow}$$

$$\sum_{i' \in F_k^{\searrow}(t_i^{\nearrow})} x_{i'j} - \sum_{i' \in F_k^{\nearrow}(t_i^{\nearrow})} x_{i'j} \ge x_{ij} - 1$$
$$\forall j \in P, \ k = a_j^0, \ \forall i \in F_k^{\nearrow}$$

$$z_{ijd} + x_{i'j} \le 1, \ \forall i \in F, \ i' \in F(d, i), \ j \in P, \ d \in D \tag{8}$$

$$x_{ij} \ge z_{ijd}, \ \forall i \in F, \ j \in P, \ d \in D \tag{9}$$

$$\sum_{i \in F_m^{\nearrow}} \sum_{j \in P} z_{ijd} \le mcap_{md} \quad \forall d \in D, \ \forall m \in MA \tag{10}$$

$$\sum_{i \in F_m^{\nearrow}} z_{ijd} = y_{jd} \quad \forall d \in D, \ \forall m \in MA, \ \forall j \in P \tag{11}$$

$$\sum_{r \in [d, d']} y_{jr} \ge 1, \quad \forall d, d' \in D / d' - d = dmax - 1, \ \forall j \in P \tag{12}$$

$$\begin{aligned} \sum_{i \in F_{d+1,d'}} x_{ij} t_{ij} \le Tmax + M_{dd'}(2 - y_{jd} - y_{jd'}) \\ + M_{dd'} \sum_{r \in [d+1, d'-1]} y_{jr} \\ \forall d, d' \in D \, s.t.: 2 \le d' - d \\ \le dmax - 1, \ d < n_d, \ \forall j \in P \end{aligned} \tag{13}$$

$$\begin{aligned} x_{ij} \in \{0, 1\}, \ \forall i \in F, \ j \in P \\ z_{ijd} \in \{0, 1\}, \ \forall i \in F, \ j \in P, \ d \in D \end{aligned} \tag{14}$$

$$y_{jd} \in \{0, 1\}, \ j \in P, \ d \in D \tag{15}$$

The objective function (7) represents the total cost of assigning airplanes to flights and assigning night maintenance activities to airplanes. Constraints (8) are used to express the fact that flight i is the last flight of the day $d = d_i^{\nearrow}$ and in its arrival airport airplane j undergo a maintenance when $y_{jd} = 1$. They impose the fact that there is no flight i' assigned to airplane j and departing after the arrival time of flight i on day d. Constraints (9) state that if a maintenance activity is assigned to airplane j in the arrival airport of flight i and on day d, necessarily flight i should be assigned to airplane j. Capacity constraints (10) guarantee that the sum of maintenance activities in a maintenance airport on each day does not exceed its maintenance capacity per day. Constraints (11) are used to express the relation between variables y and z. Constraints (12) impose that for every period of time of *dmax* consecutive days we insure that at least one maintenance has been done for each airplane. Constraints (13) represent the cumulative flight time constraints. The maximum allowed flight time between two successive maintenances is equal to *Tmax*. In expressing constraint (13) we use the fact that a given pair of days d and d' such that $1 <= d' - d <= dmax - 1$ correspond to two successive maintenance activities for aircraft j if the three conditions: $y_{jd} = 1$, $y_{jd'} = 1$ and $\sum_{r \in [d+1, d'-1]} y_{jr} = 0$ hold. The constants $M_{dd'}$ are chosen big enough to express the fact that when these conditions are not simultaneously fulfilled, no limit on cumulative flight time should be imposed. Constraints (14) and (15) impose binary conditions to variables y and z.

We note that the proposed extension to our basic Tail Assignment model is only moderately increased in size: in all the instances solved (see Tables 10 and 11) the number of variables and constraints is augmented by a factor less than 2.2.

Table 10
Computational results with maintenance constraints ($|P|$ = number of airplanes, $|H|$ = horizon (number of days), $|F|$ = number of flights).

				$dmax = 4$ days and $Tmax = 64$ hours					$dmax = 5$ days and $Tmax = 72$ hours				
$\|P\|$	$\|H\|$	$\|F\|$	# Var	# Const	# Nodes	Residual gap (%)	Gap at root (%)	CPU (sec)	# Const	# Nodes	Residual gap (%)	Gap at root (%)	CPU (sec)
10	15	417	4708	7134	2468	0	0.47	908	6688	2350	0	0.45	865
	21	496	6002	8625	4673	0	0.32	1756	8086	4450	0	0.34	1672
	30	574	7555	9813	5775	0	0.51	2578	9200	5500	0	0.56	2455
20	15	734	17,174	32,866	4599	0	0.48	3376	30,812	4380	0	0.41	3215
	21	881	21,746	39,482	6899	0	0.37	5670	37,014	6570	0	0.37	5400
	30	999	26,574	44,755	6851	0	0.39	6867	41,958	6525	0	0.39	6540
30	15	1021	35,678	67,529	8245	0	0.61	8054	63,308	7852	0	0.56	7670
	21	1178	43,388	77,899	9009	0	0.55	10,800	73,030	8580	0	0.57	9700
	30	1251	49,692	82,739	9755	0.31	0.62	10,800	77,568	9290	0.20	0.61	10,800
40	15	1225	56,962	107,172	10,710	0.30	0.70	10,800	100,474	10,200	0.23	0.66	10,800
	21	1306	64,066	114,266	10,841	0.40	0.68	10,800	107,124	10,325	0.31	0.65	10,800
	30	1494	78,892	130,684	10,873	0.46	0.65	10,800	122,516	10,355	0.40	0.66	10,800

Table 11
Computational results with maximum number of landings $\nu = 14$ and maximum spacing of 4 days ($dmax = 4$)($|P|$ = number of airplanes, $|H|$ = horizon (number of days), $|F|$ = number of flights).

$\|P\|$	$\|H\|$	$\|F\|$	# variables	# constraints	# nodes	Residual gap (%)	Gap at root (%)	CPU (sec)
10	15	417	4708	7134	2446	0	0.45	878
	21	496	6002	8625	4651	0	0.33	1534
	30	574	7555	9813	5562	0	0.47	2557
20	15	734	17,174	32,866	4570	0	0.52	3124
	21	881	21,746	39,482	6715	0	0.36	5564
	30	999	26,574	44,755	7122	0	0.36	6031
30	15	1021	35,678	67,529	8024	0	0.52	7985
	21	1178	43,388	77,899	9026	0	0.46	10,800
	30	1251	49,692	82,739	9664	0.31	0.66	10,800
40	15	1225	56,962	107,172	10,655	0.30	0.65	10,800
	21	1306	64,066	114,266	10,802	0.40	0.73	10,800
	30	1494	78,892	130,684	10,655	0.46	0.75	10,800

5.3. Computational experiments

In order to validate the proposed extension, we have carried out systematic computational experiments reported in Tables 10 and 11 below. In these experiments we have generated instances with the highest density ($\delta = 1$) imposing that for each flight timetable we have; (1) 1/4 of airports could be maintenance airports. These airports represent the most important ones based on the number of take offs and landings, (2) maintenance costs have been generated randomly.

For each pair of days (d, d') with $1 <= d' - d <= dmax - 1$, the constant $M_{dd'}$ has been calculated as follows. For each pair of flights (i, i') where i takes place on day d and i' on day d', that sequence of flights starting with i and ending with i' and satisfying the equipment continuity condition (see Section 1) which leads to maximum total flight time MFT (i, i') is determined. $M_{dd'}$ is then set equal to the maximum of MFT (i, i') over all possible pairs (i, i'). Please note that computing all the MFT (i, i') values for a given pair of days (d, d') can be done through application of a longest path algorithm in the (circuitless) graph representing all feasible sequences of flights (the complexity of such an algorithm is known to be linear in the number of arcs of the graph, therefore the total time taken by the preprocessing task of computing the $M_{dd'}$ values is practically negligible as compared with the task of solving the Tail Assignment model).

The additional constraints (8–15) have been implemented with Python 2.7 and Cplex Studio 12.6.2 All computations have been performed on a workstation running Intel(R) Core(TM) i7-3740QM CPU @ 2.7 gigahertz. For these experiments the presolve option has been activated. Tests have been carried out on 240 instances generated randomly using our data-generation procedure. Each instance corresponds to a timetable of h consecutive days for np airplanes. The various instances considered in Table 10 have been obtained by varying the value of the three parameters:

- the $dmax$ and $Tmax$ parameters either take the values $dmax = 4$ *and* $Tmax = 64$ hours or $dmax = 5$ *and* $Tmax = 72$ hours.
- the number of airplanes np: 10, 20, 30, 40;
- the time horizon h: 7, 15, 21, 30 days.

Table 11 reports on computational results obtained when imposing a maximum number ν of landings for any airplane between any two successive maintenance checks. This constraint is a special case of the cumulative constraint (13) simply obtained by replacing $Tmax$ by ν and the sum of flight durations by the total number of flights operated. In the reported experiments, the value of the ν parameter is set to 14, with $dmax = 4$ (thus a maximum of 14 landings is allowed between two consecutive maintenance checks).

The results reported confirm the efficient solvability of the model when applied to large practical-size instances, and its capability of handling all the constraints relating to Type A maintenance checks. Also, comparison with the results obtained in section 4 on the basic version of the model shows that handling maintenance constraints tends to make the problem somewhat more difficult to solve exactly, and, more precisely:

- the CPU time is about twice the one required by the basic model, and the largest instances (typically those featuring more than 1200 flight legs) are not solved to exact optimality within

the time limit of 3 hours (though the residual gaps obtained are quite reduced, typically less than 0.5%);
- significantly more nodes (typically several thousands) have to be explored in the tree search process;
- the values of the integrality gaps ('gap at root') tend to be somewhat larger than those observed for the basic model (Section 4) but are still remarkably low for such large scale MIP models (featuring up to 78,000 variables and 130,000 constraints). This confirms that the strength of the model does not significantly deteriorate when including maintenance constraints.

6. Conclusions and perspectives

In this paper, a new compact formulation for the tail assignment problem has been investigated. The computational results obtained have confirmed that, with the proposed model, real-size instances can be efficiently solved to exact optimality via current state-of-the art MIP solvers within CPU times compatible with operational use. All instances generated for the experiments can be used for benchmarks on other tail assignment model proposition. The instances and their solutions are available at http://www.lgi.ecp.fr/Tail-assignment.

The consideration of typical maintenance checks in an aircraft maintenance scheduling has been shown to be a critical one for any airline operation. We present an extension to the proposed tail assignment model taking into account 4 and 5-day maintenance, together with cumulative flight time constraints. The results obtained show that, in spite of an increase in the computational effort required, the proposed model can be used to produce exact optimal solutions to instances of size typical of practical applications. To the best of our knowledge, these results correspond to the largest instances of Tail Assignment solved to date, given that: (a) flight legs and maintenance tasks are assigned to airplanes on an individual basis; (b) all the daily maintenance constraints arising from the current regulation are simultaneously taken into account.

Among the possible topics for future research, we mention the potential interest in further investigating the links between the model of the present paper and time-space network based formulations. New ideas possibly leading to model strengthening and improved computational efficiency could emerge from such an analysis. Also worth mentioning is the investigation of *integrated models* capable of simultaneously handling aircraft *fleeting* and *routing*, and, if possible, part of *crew scheduling*. The fact that the main available techniques for addressing aircraft routing almost exclusively relied, up to now, on column generation, has certainly been a nontrivial obstacle in view of such an integration. We hope, however, that the compact model presented here will turn out to be a useful tool for eventually bypassing these difficulties, thus opening new perspectives towards increasing the efficiency of airline planning processes.

Acknowledgments

The authors thank the Associate Editor and three anonymous referees for many constructive comments that lead to a significantly improved final version of the paper.

References

Barnhart, C., Belobaba, P., & Odoni, A. R. (2003). Applications of operations research in the air transport industry. *Transportation science, 37*(4), 368–391.

Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., & Shenoi, R. G. (1998). Flight string models for aircraft fleeting and routing. *Transportation Science, 32*(3), 208–220. doi:10.1287/trsc.32.3.208.

Borndörfer, R., Dovica, I., Nowak, I., & Schickinger, T. (2010). Robust tail assignment. *Technical Report*. Konrad-Zuse-Zentrum fur Informationstechnik Berlin.

Clarke, L., Johnson, E., Nemhauser, G., & Zhu, Z. (1997). The aircraft rotation problem. *Annals of Operations Research, 69*(0), 33–46. doi:10.1023/A:1018945415148.

Clarke, L. W., Hane, C. A., Johnson, E. L., & Nemhauser, G. L. (1996). Maintenance and crew considerations in fleet assignment. *Transportation Science, 30*(3), 249–260.

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.

Daskin, M. S., & Panayotopoulos, N. D. (1989). A Lagrangian relaxation approach to assigning aircraft to routes in hub and spoke networks. *Transportation Science, 23*(2), 91–99.

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

Froyland, G., Maher, S. J., & Wu, C.-L. (2013). The recoverable robust tail assignment problem. *Transportation Science, 48*(3), 351–372.

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

Grönkvist, M. (2005). *The tail assignment problem*. Chalmers, Göteborg University (Ph.D. thesis).

Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., & Sigismondi, G. (1995). The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming, 70*(1-3), 211–232.

Haouari, M., Shao, S., & Sherali, H. D. (2012). A lifted compact formulation for the daily aircraft maintenance routing problem. *Transportation Science, 47*(4), 508–525.

Jünger, M., Elf, M., & Kaibel, V. (2003). Rotation planning for the continental service of a european airline. In *Mathematics key technology for the future* (pp. 675–689). Springer.

Kabbani, N. M., & Patty, B. W. (1992). Aircraft routing at American airlines.. In *Proceedings of the AGIFORS symposium*.

Klabjan, D. (2005). Large-scale models in the airline industry. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column generation* (pp. 163–195). Springer US.

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

Lavoie, S., Minoux, M., & Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research, 35*(1), 45–58.

Levin, A. (1971). Scheduling and fleet routing models for transportation systems. *Transportation Science, 5*(3), 232–255.

Liang, Z., & Chaovalitwongse, W. A. (2012). A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem. *Transportation Science, 47*(4), 493–507.

Liang, Z., Chaovalitwongse, W. A., Huang, H. C., & Johnson, E. L. (2011). On a new rotation tour network model for aircraft maintenance routing problem. *Transportation Science, 45*(1), 109–120.

Mercier, A., Cordeau, J.-F., & Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research, 32*(6), 1451–1476. http://www.sciencedirect.com/science/article/pii/S0305054803003447, doi: 10.1016/j.cor.2003.11.013.

Minoux, M. (1984). Column generation techniques in combinatorial optimization, a new application to crew pairing problems. Proceedings of the twenty-fourth AGIFORS Symposium, Strasbourg, France.

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.

Sherali, H. D., Bish, E. K., & Zhu, X. (2006). Airline fleet assignment concepts, models, and algorithms. *European Journal of Operational Research, 172*(1), 1–30.