Journal of Air Transport Management 16 (2010) 258–263

Contents lists available at ScienceDirect

Journal of Air Transport Management

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

Aircraft routing under different business processes

Eve Lacasse-Guay*, Guy Desaulniers, François Soumis

École Polytechnique de Montréal and GERAD, Department of Mathematics and Industrial Engineering, C.P. 6079, Succ. Centre-Ville, Montréal, Québec H3C 3A7, Canada

Keywords:
Aircraft routing
Maintenance requirements
Strings
One-day routes

ABSTRACT

Given a set of scheduled flights that must be operated by the same aircraft type, the aircraft routing problem consists of building anonymous aircraft routes that respect maintenance requirements and cover each flight exactly once. This paper looks at the nature of the problem and introduces a classification according to three business processes that are used to assign the anonymous routes to the specific aircraft tail numbers. Furthermore, we compare the aircraft routing problem variants resulting from these three processes with regard to their adaptability to different contexts, the difficulty of solving them, the cost of the computed solutions, and the robustness of these solutions.

1. Introduction

Major commercial airlines rely on a sequential process to plan their operations for a given season. Typically, this process consists of solving a flight scheduling problem to determine the schedule to operate, a fleet assignment problem to assign an aircraft type to each flight, an aircraft routing problem to build aircraft routes that meet maintenance requirements, and a crew scheduling problem to determine work schedules for the pilots and flight attendants[1].

We focus on the aircraft routing problem (ARP) which is separable by aircraft type. Thus, we assume for the rest of this paper that there is a single aircraft type. Given a set of flight legs to be operated by this aircraft type, the ARP consists of determining feasible aircraft routes such that each leg is covered exactly once by an aircraft and the number of aircraft available is not exceeded. An aircraft route is a sequence of flight legs, connections, and maintenance checks. It is feasible if, for any pair (l_1, l_2) of consecutive legs in this route, the arrival station of l_1 is the same as the departure station of l_2, the duration between the arrival time of l_1 and the departure time of l_2 exceeds the minimal connection time at the connecting station, and this duration exceeds the minimal maintenance time (that includes the minimal connection time) if a maintenance check is scheduled between l_1 and l_2. Moreover, an aircraft route must contain a maintenance check at most at every δ calendar days (δ can depend on the aircraft type) and all maintenance checks must be scheduled at a station where maintenance can be performed for the given aircraft type. In certain cases, checks can be performed at any time, while in others, they must be realized during the night.

In general, only maintenance checks of type A are considered when solving the ARP. According to safety regulations, the frequency of these checks depends on the number of flying hours and takeoffs, instead of a number of calendar days. However, to ensure robustness in their operations, airlines often use a number of days that is more restrictive than the number of flying hours and takeoffs. Typically, δ is equal to 3 or 4 and a check lasts approximately 4 to 5 h. Other maintenance types must also be performed on each aircraft. Type B maintenances must occur approximately every month, while type C and D maintenances are required a few times per year. Because of their low frequencies, these maintenance types are usually not taken into account in the ARP.

The ARP can be defined over a cyclic or an acyclic (dated) horizon. Indeed, when planning the aircraft routes for a whole season, the overall solution process is often divided into two phases. The first phase considers a typical schedule for a week (or a day depending on the schedule regularity) and computes a cyclic solution that can be duplicated week after week. Since all the aircraft are identical, a cyclic solution is feasible if each weekly route that it yields can be performed week after week by any of the aircraft, not necessarily the same aircraft. In the second phase, the duplicated solution is reoptimized over dated horizons that contain exceptions to the typical schedule (for instance, over each holiday weekend). In this case, the problem to solve for each dated horizon includes initial conditions (initial location and number of days since last maintenance for each aircraft) and similar final conditions. This two-phase process favors regular aircraft connections, which eases crew scheduling and ground services.

At this planning stage, the routes are built for anonymous aircraft. Close to the day of operations, these routes are assigned to

* Corresponding author.
E-mail address: eve.lacasse-guay@gerad.ca (E. Lacasse-Guay).

[1] See Klabjan, 2005, for a detailed description of these problems.

0969-6997/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jairtraman.2010.02.001

the aircraft tail numbers. In fact, they are often revised (almost on a daily basis) to take into account the perturbations (flight delays or cancellations, aircraft mechanical failures, etc.) that can occur during the operations. This also provides the opportunity to schedule B, C and D maintenance checks for specific aircraft. This paper is not concerned with this aircraft assignment problem. It focuses on the ARP for anonymous aircraft.

The ARP can be treated as a feasibility problem because operational costs of anonymous aircraft of the same type are identical. However, various objectives can also be considered, solely or in combination, such as minimizing the number of aircraft used, maximizing solution robustness by avoiding short connections as much as possible, and maximizing the number of connections that are attractive for the passengers (also referred to as maximizing through values).

In the industry, there exist three main business processes that govern how aircraft tail numbers are assigned to the planned aircraft routes. These processes influence the ARP definition, yielding three variants of the ARP:

- Strings: Each aircraft is assigned to a string after each maintenance check. A string is a sequence of connected flights that starts just after a maintenance check and ends with the next maintenance check. It can last up to δ days. When several aircraft complete a maintenance check in a same station at approximately the same time (especially when they remain overnight), they can all be assigned to the different strings beginning soon after, providing some flexibility for assigning the aircraft.
- Big cycle: Assuming a completely regular schedule over a cyclic daily horizon, the goal of this business process is to ensure equal aircraft utilization. It imposes that the planned routes form a big cycle that covers each flight exactly once. All aircraft are then assigned to this cycle with a one-day delay between each aircraft. The route of each specific aircraft is thus fixed until the end of the validity period of the regular schedule.
- One-day routes: Assuming that maintenance can only be performed overnight, each aircraft is assigned each night to a route that lasts one day. During the planning stage, these one-day routes are built in such a way that there is a high probability that each aircraft can reach a maintenance station if needed.

The three ARP variants derived from these business processes are denoted ARP-S, ARP-BC, and ARP-ODR, respectively.

Some papers on the ARP assume that there is no need to plan the type A maintenance checks because they can be planned easily just before the operations. This can be the case for a hub-and-spoke network in which every aircraft frequently visits a hub where it can be maintained or for relatively small airlines whose aircraft always remain overnight in a few stations equipped for maintenance checks[2].

2. Strings

Here we consider the ARP-S variant resulting from the business process using strings. The main goal of the ARP-S is to determine strings that cover every flight in the schedule exactly once. In addition, the ARP-S must take into account constraints ensuring that strings can be concatenated to form routes spanning the complete horizon and satisfying aircraft availability. Note that the maintenance requirements are directly imposed in the string definition.

The notion of string was introduced by Barnhart et al. (1998a) who consider the fleet assignment problem and the ARP-S over a cyclic horizon simultaneously. They formulate this combined problem as an integer program that involves a binary variable for each pair of aircraft type and string that can be assigned to this type. To solve this model, they develop a branch-and-price method (Barnhart et al., 1998b; Desaulniers et al., 1998; Lübbecke and Desrosiers, 2005) that incorporates different heuristic acceleration techniques. With this method, they solve in 5.5 h an instance involving 9 aircraft types and 1124 long-haul flights over a weekly horizon. Their method can also be adapted to the case where the maintenance requirements are expressed as an upper bound on the number of flying hours between two consecutive maintenances. In that case, the column generation sub problem becomes a resource-constrained shortest path problem (Irnich and Desaulniers, 2005). This case was tested on several ARP-S data sets involving a single aircraft type and up to 190 short-haul flights from a daily schedule. On the largest instance, their method required close to 4 h. The proposed approach can easily be modified to consider an acyclic horizon, although this is not mentioned in their paper.

Recently, Lan et al. (2006) revisited the approach of Barnhart et al. (1998a) with the aim of selecting strings that are less prone to the propagation of passenger delays in case of a perturbation of the planned operations. The new model, which includes a stochastic objective function, is also solved by a branch-and-price method that was tested on small-sized instances.

Other papers have addressed the ARP-S or variants of it without explicitly using strings. In these works, routes made up of strings are directly built. Feo and Bard (1989) consider a strategic ARP-S where the locations of the maintenance stations are also decision variables. They assume that one-day routes covering all scheduled flights of a cyclic weekly horizon are given and fixed, and that maintenances can only be performed at night. To solve this problem, they propose a two-phase heuristic based on a set covering formulation.

Sriram and Haghani (2003) combined fleet assignment and ARP-S problem over a cyclic weekly horizon and, as in Feo and Bard, assume that one-day routes are given and maintenances can occur only at night. The combined problem is formulated as a multi-commodity network flow model with side constraints that is restrictive with regards to the cyclicity of the solution. Indeed, it imposes that each aircraft performs the same exact one-week route week after week in the duplicated solution. They develop a heuristic solution method based on a random and a depth-first search, and compare it with a commercial branch-and-boundmethod. Sriram and Haghani also extend their multi-commodity network flow model to the case where maintenance requirements are expressed as an upper bound on the number of flying hours between two maintenances.

Finally, Lacasse-Guay (2009) introduces a multi-commodity network flow model with side constraints for the ARP-S overa cyclic weekly horizon. In this model that aims at building one-week routes, variables indicating how these routes connect from one week to the next are used to force the maintenance requirements in a cyclic fashion. For small-sized instances, this model can be solved to optimality using a commercial solver. For larger instances, the author develops a heuristic branch-and-bound algorithm that performs similarly to her implementation of the heuristic branch-and-price method of Barnhart et al. (1998a).

3. Big cycle

The ARP-BC ensues from the big cycle business process. It is defined for a completely regular flight schedule over a cyclic daily

[2] This work includes Afsar et al. (2006) and Haouari et al. (2009) that focus on routing the aircraft to meet scheduled long-term maintenance checks over an acyclic horizon, and Elf et al. (2003) that concentrates on determining routes that minimize the risk of delays during the operations.

horizon and differs from the ARP-S only by the requirement that the strings can be concatenated to form a cycle respecting the maintenance requirements and covering each flight exactly once.

Clarke et al. (1997) model the ARP-BC as a network flow model with side constraints. In the underlying network, the arc set consists of flight arcs and waiting arcs, while nodes represent flight departures and arrivals. Maintenance opportunities are included in waiting arcs associated with a maintenance station and a sufficiently long idle time. A restrictive assumption made by Clarke et al. is that the network comprises only the waiting arcs used in the fleet assignment solution. To justify this restriction, the authors stipulate that violating these waiting arcs may result in the need for additional aircraft. Consequently, finding a big cycle in this network is equivalent to finding an Eulerian tour that covers every arc exactly once and satisfies the maintenance constraints. Based on this observation, Clarke et al. propose an integer program for the ARP-BC that includes subtour elimination constraints and infeasible path constraints. To solve it, they apply a Lagrangian heuristic that adds these constraints as they are violated. The use of infeasible path constraints allows them to be more specific about the maintenance requirements. Indeed, they consider that every aircraft must perform a routine maintenance check at least every three days and an avionics maintenance check at least every four days. With the proposed method, the largest instance solved (in less than 1 h) involves 3818 arcs and 1095 nodes (corresponding to approximately 2500 flights).

Barnhart et al. (1998a) present a daily version of their string model that contains additional constraints, called connectivity constraints, to ensure that the strings can be concatenated to form a big cycle. These constraints are equivalent to subtour elimination constraints. The resulting model is solved using branch-and-price combined with constraint generation. With this methodology, they solve a daily ARP-BC instance involving 190 flights in approximately 10 h.

Gopalan and Talluri (1998) consider the ARP-BC in the case where maintenance must be performed on each aircraft at least every three days during the night. For this problem, they develop an iterative two-phase method that builds one-day routes in the first phase and determines a cycle satisfying the maintenance requirements in the second phase. When no such cycle can be found, another iteration is started by modifying heuristically the one-day routes from the first phase (this step can include a change of aircraft type for certain flights). The second phase consists of finding an Eulerian tour that covers each one-day route once and satisfies the three-day maintenance constraint. To do so, the authors develop a polynomial-time algorithm. Under a k-day maintenance constraint with $k \geq 4$, this second-phase problem would be NP-complete (Talluri, 1998).

4. One-day routes

In the ARP-S and ARP-BC variants, routes are built without taking into account the (relatively high) probability of disruptions during the operations. The philosophy behind the ARP-ODR differs significantly as it considers the ARP as a stochastic optimization problem. Given that disruptions are frequent, route reassignment is performed on a daily basis, and maintenance occurs only at night, some airlines prefer to plan only one-day routes that are not concatenated to form complete routes spanning the whole horizon. The one-day routes must, however, be built in such a way that there are very high chances that all the aircraft can meet their maintenance requirements. In other words, they must be such that, for each station where aircraft remain overnight, there are enough routes beginning at this station and ending in a maintenance station. When solving the ARP-ODR for a daily flight schedule, this can be achieved by maximizing the expectation that all aircraft requiring a maintenance check at the end of the day are assigned to a route ending at a maintenance station, assuming that a distribution of the probabilities that an aircraft needs such a maintenance is given.

The ARP-ODR can be applied over an acyclic or a cyclic horizon. Assuming that the probability distributions are independent from one day to another, the problem becomes separable by day of the horizon. Furthermore, if the flight schedule is the same for several days of the horizon (except the first and the last day in an acyclic context), then a single one-day problem needs to be solved for all these days.

In the industry, the one-day routes business process is a recent trend among some low-cost air carriers. Consequently, literature is rare on the ARP-ODR (Heinold, 2008) that was conducted at Southwest airlines. He studies an ARP-ODR over a cyclic horizon where he considers an identical flight schedule for each day of the horizon. He models the problem as a stochastic integer nonlinear program that allows changing the aircraft type on certain flights. Nonlinearity appears only in the objective function. The author proposes to solve this program with a successive linearization method using a commercial solver. With this method, he tackled a one-day instance involving around 3460 flights.

5. Business process comparison

5.1. Application contexts

The ARP variants have been tackled in previous studies for varying application contexts. These contexts might differ due to the underlying business process, or because the proposed solution methodologies are not suitable for all contexts.

The ARP-S can be applied for an acyclic or a cyclic horizon and for an irregular flight schedule that can change from one day to another. Although some papers assume that maintenance checks can only be performed at night (which is common practice in several airlines), the papers of Barnhart et al. (1998a) and Lan et al. (2006) use a branch-and-price method that allows maintenance at any time during the day. This methodology also offers the possibility to express the maintenance requirements as an upper bound on the number of flying hours between two consecutive maintenance checks. The ARP-S was combined with the fleet assignment problem.

Given that the main goal of the big cycle business process is to ensure equal aircraft utilization, the ARP-BC has been defined for quite restrictive contexts that simplify the achievement of this goal. Indeed, the ARP-BC requires a cyclic daily horizon and a completely regular flight schedule (this case is rare in practice due to weekend days where the flight schedule usually differs from that of a week day). On the other hand, the solution method of Barnhart et al. (1998a) can handle, as above, maintenance checks at any time and maintenance requirements in a maximum number of flying hours. These issues can also be treated in the Lagrangian heuristic of Clarke et al. (1997) due to the use of infeasible path constraints. In fact, such constraints can be used for dealing with a wide variety of maintenance feasibility rules.

The ARP-ODR is applicable for an acyclic or a cyclic horizon, and also for an irregular flight schedule. It can be combined with fleet assignment. It requires that maintenance checks be performed only at night and that the maintenance requirements be expressed as a maximum number of days between two maintenances.

In summary, the ARP-S has the widest range of applicability. Both ARP-BC and ARP-ODR variants have restrictions that differ. The ARP-BC is restrictive with respect to the horizon type and the schedule regularity, while the ARP-ODR requires restrictive maintenance rules (which are, however, commonly used in the industry for short- and medium-haul fleets).

5.2. Computational requirements

Comparing computational requirements of different problems is not an easy task because the methods for solving them might differ considerably and they have been tested, in general, on different instances using different computers. Nevertheless, we want to emphasize certain characteristics of the ARP variants and their solution methods that influence the computational times. This will allow to roughly rank the variants with regards to their computational requirements. To do this comparison, let us consider an instance defined over a cyclic weekly horizon.

First, assume that the flight schedule is not regular from one day to another. In this case, the ARP-BC is not applicable. For the ARP-S, all the flights in the schedule must be considered at once. This is not the case for the ARP-ODR which is separable per day of the week. This gives a significant advantage for the ARP-ODR in terms of computational times. To reduce computational times for the ARP-S, some researchers (see Feo and Bard, 1989, and Sriram and Haghani, 2003) have assumed that one-day routes were given as an input. This reduces considerably the computational times but yields, most probably, more expensive solutions.

Next, assume a regular daily flight schedule, still over a weekly horizon. In this case, the ARP-BC is applicable and requires solving a single one-day problem. Again, the ARP-ODR is separable per day. Given the regular schedule, only one such one-day problem needs to be solved. Because the methods of Barnhart et al. (1998a) and Clarke et al. for solving the ARP-BC (which likely produces better quality solutions than the two-phase heuristic of Gopalan and Talluri, 1998) require a constraint generation procedure to add subtour elimination constraints, the ARP-ODR seems easier to solve than the ARP-BC. On the other hand, it is obvious that the ARP-S is computationally expensive when the whole horizon is considered at once. Nevertheless, its computational performance can be improved by first solving a daily problem and second duplicating its solution over the week (as realized for the ARP-BC). In this case, the computational requirements would be slightly higher than those of the ARP-BC but the quality of the computed solutions would diminish.

From this discussion, we propose to rank the ARP variants as follows: the ARP-S is the most computationally extensive variant, followed by the ARP-BC. The ARP-ODR is the easiest variant to solve.

5.3. Solution costs

We compare the costs of the solutions produced by the different ARP variants. To do so, we consider, without loss of generality, a weekly cyclic horizon with a regular daily flight schedule. First, we justify why the relations

$$D_{\text{ARP}-\text{BC}} \subseteq D_{\text{ARP}-\text{Sdaily}} \subseteq D_{\text{ARP}-\text{Sweekly}} \tag{1}$$

hold, where D_i indicates the feasible domain of problem i. ARP-S_{daily} indicates that the ARP-S is solved on a daily version of the problem and its solution is duplicated over the weekly horizon, while ARP-S_{weekly} corresponds to solving the ARP-S directly on the whole horizon. In general, these inclusions are strict, showing that the ARP-S_{weekly} can yield less costly solutions than the ARP-S_{daily}, that can produce itself less costly solutions than the ARP-BC. Second, we provide an example where the ARP-S_{daily} is feasible while the ARP-BC is infeasible and another example where there exists an ARP-S_{weekly} solution that uses less aircraft than any ARP-S_{daily} solution. Finally, we discuss the ARP-ODR separately.

5.3.1. ARP-BC vs ARP-S_{daily}

The first inclusion ($D_{\text{ARP-BC}} \subseteq$ D_ARP-S_{daily}) comes from the fact that the only difference between the ARP-BC and the ARP-S_{daily} is the addition of a constraint imposing that the strings form a cycle in the ARP-BC. The following small example shows that this inclusion can be strict, that is, there can exist feasible solutions to the ARP-S_{daily} that are not feasible for the ARP-BC.

This example considers a daily schedule involving six flights, numbered from 1 to 6, linking five stations A, B1, B2, C and D. One-day flights are used to develop a simpler example. Assume that six aircraft are available, stations B1 and B2 are the maintenance stations, and maintenance checks must be performed there at most at every $\delta = 3$ days. Fig. 1 illustrates a feasible solution for the ARP-S_{daily}. The first string, represented by a plain line in the figure, begins in station B1, executes flight sequence 2-1-3 and ends in station B1. The second string, represented by a dotted line, begins in B2, executes flights 5-4-6 and ends in station B2. These two strings only have one station in common: they meet in station A. This ARP-S_{daily} solution does not satisfy the big cycle constraint.

It is possible to create a cycle in this example by swapping the aircraft in station A. However, the string composed of the flights 5-3 lasts two days, while the string 2-1-4-6 does not respect the maintenance rule as it lasts four days. Consequently, this solution is not maintenance-feasible even if the network is Eulerian. In this case, the planners would need to revise the fleet assignment to ensure that a feasible ARP-BC solution exists. Such a revision usually decreases the expected profits.

5.3.2. ARP-S_{daily} vs ARP-S_{weekly}

The second inclusion ($D_{\text{ARP-Sdaily}} \subseteq D_{\text{ARP-Sweekly}}$) results from an implicit constraint added to the daily problem: the same flight cannot be repeated in a string, i.e., a string lasting more than one

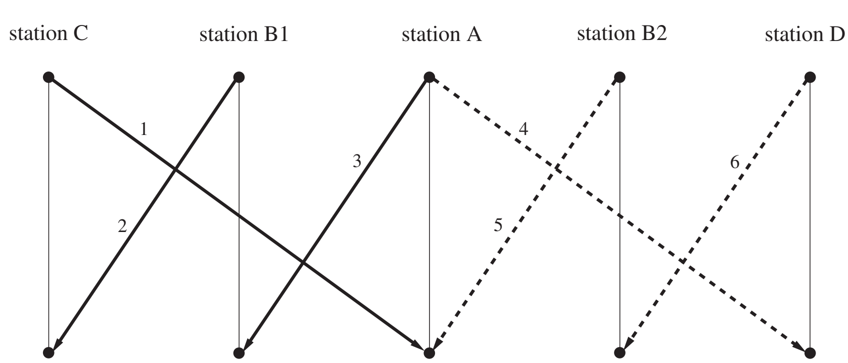

Fig. 1. A two-string solution for the ARP-S_{daily}.

day cannot contain a flight f on its first day and on its second day for instance. Otherwise, the duplicated solution would cover f twice on both of these days. Given this implicit constraint, it is easy to observe that this inclusion is strict in many cases.

The following example shows that, for certain instances, the ARP-S$_{\text{daily}}$ approach can yield solutions requiring more aircraft than in a solution obtained by the ARP-S$_{\text{weekly}}$ approach. Assume that maintenance checks must be performed at most at every $\delta = 3$ days. Consider a network composed of two similar subnetworks G and G' (see Fig. 2). The subnetwork G involves four stations (A, B, C, D) and eight flights, numbered from 1 to 8, as illustrated in Fig. 2(a). The subnetwork G', illustrated in Fig. 2(b), is very similar as it contains also four stations (A′, B, C′, D′) and eight similar flights (1′-8′). Station B is the unique maintenance station, which is common to both subnetworks. There are no flights between the stations A, C and D and the stations A′, C′ and D′. In this figure, the M arcs indicate the maintenance opportunities upon an arrival at the maintenance station and their durations. In particular, the M arc adjacent to the flight 1 arc shows that an aircraft going to maintenance after flight 1 cannot connect to flight 3 unless it waits almost a complete day.

For the ARP-S$_{\text{daily}}$, one can observe that any string covering flight 7 must include flights 3 and 4 because a string must start and end in B. Therefore, such a string lasts at least two days. The most productive of them, 3-7-8-5-6-4, lasts exactly two days and, thus, requires two aircraft in the overall solution. A similar reasoning for flight 7′ yields another string (3′-7′-8′-5′-6′-4′) that lasts two days. Consequently, two other aircraft are needed. Next, remark that flight 2 (resp. 2′) must be followed by flight 1 (resp. 1′). Consequently, a feasible solution to ARP-S$_{\text{daily}}$ concatenates the four flight sequences 3-7-8-5-6-4, 3′-7′-8′-5′-6′-4′, 2-1, and 2′-1′ to obtain routes that satisfy the 3-day maintenance requirements. By enumeration, we can prove that all feasible solutions require at least six aircraft. An example of a six aircraft solution is:

$$3-7-8/5-6-4-2/1-M/3'-7'-8'/5'-6'-4'-2'/1'-M \qquad (2)$$

where a / represents a night and M a maintenance check. Note that this solution also forms a big cycle.

A solution feasible for the ARP-S$_{\text{weekly}}$ is given in Table 1. The upper part presents the strings for the subnetwork G. The first line contains three identical strings lasting two days: 1-3-4-2/1-M-2. Flights 1 and 2 appear only in these strings. The maintenance check is in the middle of the day and flight 2 can be operated after this check in the same day. The lines 2, 3 and 4 contain three identical three-day strings: 3-7-8/5-6-7-8/5-6-4-M. The flights 5, 6, 7 and 8 appear only in these strings. These strings for subnetwork G require two or three aircraft per day. The lower part provides the strings for the subnetwork G'. They correspond to the same strings as for subnetwork G, but they apply to the flights i' and are shifted by one day. Combining these strings we obtain a solution that uses five aircraft each day. Indeed, the three-day strings can easily be connected together as one of these strings starts at the maintenance base each morning and another one ends with a maintenance each night.

With this example, we can also observe a potential gain in the expected profits when the ARP-S$_{\text{weekly}}$ is combined with the fleet assignment problem. Assume that there are two aircraft types, I and II, and three aircraft of each type. In addition, suppose that flights 1, 2, 1′ and 2′ are more profitable with type I, flights 5, 6, 7, 8, 5′, 6′, 7′ and 8′ more profitable with type II, and flights 3, 4, 3′ and 4′ yield the same profits with both types. In this case, for the ARP-S$_{\text{weekly}}$ solution given in Table 1, it is possible to assign a type I aircraft to the flights 1, 2, 1′ and 2′, and a type II aircraft to the flights 5, 6, 7, 8, 5′, 6′, 7′ and 8′. On the other hand, this is not possible for an ARP-S$_{\text{daily}}$ solution that uses the six available aircraft as the aircraft assigned to the flights 5, 6, 7 and 8 must also operate either the flights 1 and 2 or the flights 1′ and 2′. This would also be the case for an ARP-BC solution.

Table 1
A feasible solution to the ARP-S$_{\text{weekly}}$.

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Subnetwork G	1-3-4-2	1-M-2	1-3-4-2	1-M-2	1-3-4-2	1-M-2
		3-7-8	5-6-7-8	5-6-4-M		
				3-7-8	5-6-7-8	5-6-4-M
	5-6-7-8	5-6-4-M				3-7-8
No. of aircraft in G	2	3	2	3	2	3
Subnetwork G'	1′-M-2′	1′-3′-4′-2′	1′-M-2′	1′-3′-4′-2′	1′-M-2′	1′-3′-4′-2′
	3′-7′-8′	5′-6′-7′-8′	5′-6′-4′-M			
			3′-7′-8′	5′-6′-7′-8′	5′-6′-4′-M	
	5′-6′-4′-M				3′-7′-8′	5′-6′-7′-8′
No. of aircraft in G'	3	2	3	2	3	2
Total no. of aircraft	5	5	5	5	5	5

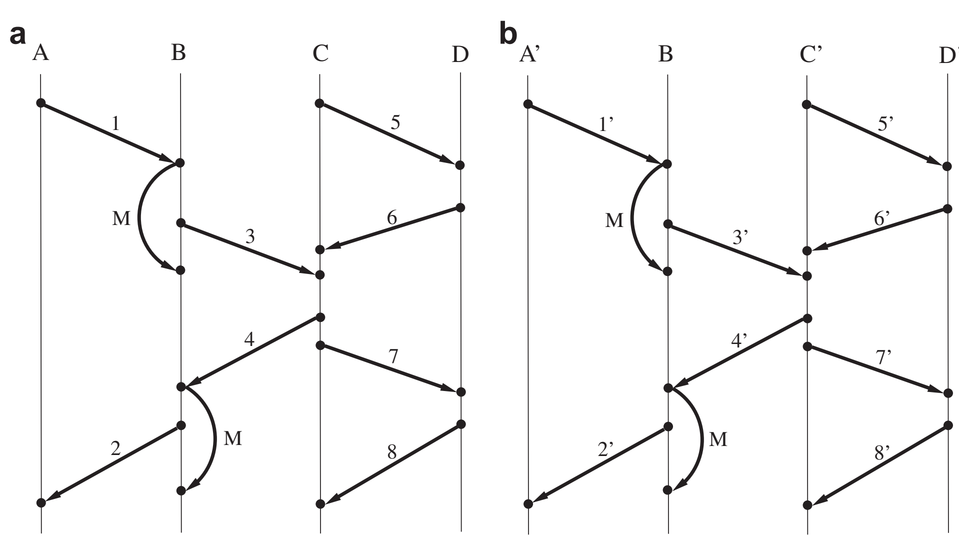

Fig. 2. A daily network composed of two subnetworks.

Table 2
Probability that a route is not disturbed in function of its length.

p	Route length (days)						
	1	3	4	7	50	100	200
0.18	0.82	0.55	0.45	0.25	5e-05	2e-09	6e-18
0.20	0.80	0.51	0.41	0.21	1e-05	2e-10	4e-20
0.22	0.78	0.47	0.37	0.18	4e-06	2e-11	3e-22
0.24	0.76	0.44	0.33	0.15	1e-06	1e-12	1e-24
0.25	0.75	0.42	0.32	0.13	6e-07	3e-13	1e-25
0.26	0.74	0.41	0.30	0.12	3e-07	8e-14	7e-27
0.28	0.72	0.37	0.27	0.10	7e-08	5e-15	3e-29
0.30	0.70	0.34	0.24	0.08	2e-08	3e-16	1e-31
0.32	0.68	0.31	0.21	0.07	4e-09	2e-17	3e-34

5.3.3. *ARP-ODR*

Because the one-day routes built in the ARP-ODR are not concatenated to form complete routes, it is difficult to compare the feasible domain of the ARP-ODR with the feasible domains of the other ARP variants. Nevertheless, we can say that $D_{\text{ARP-ODR}}$ is not restricted by the big cycle constraint and, if strings were built from the one-day routes, they would not be subject to the implicit constraint appearing in the ARP-S_{daily}. Thus, $D_{\text{ARP-ODR}}$ contains a large number of solutions that do not belong to $D_{\text{ARP-BC}}$ and $D_{\text{ARP-Sdaily}}$.

The strength of the ARP-ODR is its stochastic treatment of the problem. It aims at maximizing the expectation that the aircraft meet their maintenance requirements, even if perturbations occur during the operations. Consequently, if recovery costs for these perturbations were taken into account in the evaluation of the planned solutions, then the ARP-ODR approach would certainly produce solutions that are less costly than those obtained with the other approaches.

5.4. *Solution robustness*

Here we look at the robustness of the solutions produced by the three ARP variants with regards to the disruptions that can modify the planned routes. Given that the different business processes do not assign the routes to the aircraft similarly, the length of the routes to consider in this study depends on the business process. Indeed, the string process assigns routes lasting up to three or four days, the big cycle process assigns routes that can last several months, while the daily routes process assigns one-day routes.

Let p be the probability that a route is disrupted during a day. Assume that this probability is the same every day and that the probabilities are independent from one day to another. Then, the probability P_k that a route lasting k days is not disrupted is given by $P_k = (1\text{-}p)^k$. Table 2 provides the value of P_k for different values of p ranging between 0.18 and 0.32, and different values of k, namely, $k = 1, 3, 4, 7, 50, 100$ and 200 days.

First, we observe that, for a 7-day route, the probability that an aircraft executes the planned route is 13% when p is equal to 0.25. This is the same value as the one reported by Southwest airlines (Heinold, 2008). Thus, the values selected for p (between 0.18 and 0.32) seem reasonable.

The results of Table 2 indicate that, among the three ARP variants, the ARP-ODR produces the most robust solutions. With such solutions, it remains possible, in most cases, to assign overnight a one-day route ending at a maintenance base to every aircraft due for maintenance even if its previous one-day route was changed in response to a disruption. Indeed, the planning has optimized the probabilities that, for any given station with overnighting aircraft, there is enough one-day routes starting in this station and ending in a maintenance station.

6. Conclusion

In this paper, we have presented a survey of the literature on three ARP variants that arise under three different business processes, namely, the ARP-S, the ARP-BC, and the ARP-ODR. We also compared these variants with respect to different viewpoints. It turns out that the ARP-S is the most adaptable variant. It yields low cost solutions that are more or less robust. Its weakest point is that it is the most computationally expensive variant. For most criteria, the ARP-BC is the worst variant. Indeed, it has restricted applicability and can produce costly solutions that are far from being robust. Finally, the ARP-ODR performs the best for the computational requirements, the solution cost, and the solution robustness. On the other hand, it is subject to restrictive maintenance rules, which are however common practice in several airlines.

References

Afsar, H.M., Espinouse, M.-L., Penz, B., 2006. A two-step heuristic to build flight and maintenance planning in a rolling-horizon. In: 2006 IEEE International Conference on Service Systems and Service Management, Troyes, France, 2, 1251–1256.

Barnhart, C., Boland, N., Clarke, L.W., Johnson, E.L., Nemhauser, G.L., Shenoi, R.G., 1998a. Flight string models for aircraft fleeting and routing. Transportation Science 32, 208–220.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998b. Branch-and-price: column generation for solving huge integer programs. Operations Research 46, 316–329.

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

Desaulniers, G., Desrosiers, J., Ioachim, I., Soumis, F., Solomon, M.M., Villeneuve, D., 1998. A unified framework for time constrained vehicle routing and crew scheduling problems. In: Crainic, T.G., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, Norwell.

Elf, M., Jünger, M., Kaibel, V., 2003. Rotation planning for the continental service of a European airline. In: Jager, W., Krebs, H. (Eds.), Mathematics – Key Technologies for the Future: Joint Projects between Universities and Industry. Springer, New York.

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

Gopalan, R., Talluri, K., 1998. The aircraft maintenance routing problem. Operations Research 46, 260–271.

Haouari, M., Aissaoui, N., Mansour, F.Z., 2009. Network flow-based approaches for integrated aircraft fleeting and routing. European Journal of Operational Research 193, 591–599.

Heinold, A., 2008. Maintenance reachability at southwest airlines. In: Proceedings of the AGIFORS Scheduling and Strategic Planning Study Group Meeting, Dallas.

Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation. Springer, New-York.

Klabjan, D., 2005. Models in the airline industry. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation. Springer, New York.

Lan, S., Clarke, J.-P., Barnhart, C., 2006. Planning for robust airline operations: optimizing aircraft routings and flight departure times to minimize passenger disruptions. Transportation Science 40, 15–28.

Lacasse-Guay, E., 2009. Affectation des types d'avions aux vols avec contraintes de maintenance. In: Ph.D. dissertation, École Polytechnique de Montréal, Montréal, in French.

Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53, 1007–1023.

Sriram, C., Haghani, A., 2003. An optimization model for aircraft maintenance scheduling and re-assignment. Transportation Research Part A 37, 29–48.

Talluri, K., 1998. The four-day aircraft maintenance routing problem. Transportation Science 32, 43–53.