EURO J Transp Logist (2014) 3:121–142
DOI 10.1007/s13676-013-0038-9

RESEARCH PAPER

Airline fleet assignment with internal passenger flow reevaluations

David Lasalle Ialongo · Guy Desaulniers

Received: 27 July 2012 / Accepted: 22 October 2013 / Published online: 5 November 2013

Abstract The airline fleet assignment problem consists of assigning an aircraft type to each flight leg of a flight schedule in order to maximize the airline expected profit. Most existing fleet assignment models (FAMs) use an estimation of the revenues per flight leg that neglects the interdependency between the flight legs and poorly approximates the spill and recapture of the passengers. To overcome this difficulty, Dumas et al. (Transp Res Part B 43(4):466–475, 2009) have introduced an iterative solution method that solves at each iteration a FAM and a passenger flow model (PFM). A solution to the PFM provides the expected number of passengers on each leg, taking into account spill and recapture. These numbers are then used to better estimate the revenues per flight leg for the next iteration. Compared to solving a FAM once, this method yields better quality solutions but requires much larger computational times (by a factor 10 or more). In this paper, we aim at reducing these computational times while preserving solution quality. To do so, we propose to reevaluate periodically the flight leg revenues via the PFM while solving the FAM with a heuristic branch-and-bound algorithm. Computational results obtained for a large-scale real-life network and various demand levels show that the proposed method can reduce the average computational time by a factor of 2–3 to obtain solutions of similar quality.

Keywords Airline operations planning · Fleet assignment · Passenger flow · Estimated passenger revenues · Heuristic branch-and-bound · Variable fixing

D. Lasalle Ialongo · G. Desaulniers (✉)
Mathematics and Industrial Engineering Department, École Polytechnique de Montréal and GERAD, C.P. 6079, Succ. Centre-Ville, Montreal H3C 3A7, Canada
e-mail: Guy.Desaulniers@gerad.ca

D. Lasalle Ialongo
e-mail: david.lasalle.ialongo@gerad.ca

1 Introduction

Planning the operations of an airline is complex and usually divided into several steps (Klabjan 2005): flight schedule design, fleet assignment, aircraft scheduling including maintenance checks, and crew scheduling. The fleet assignment problem aims at maximizing the productivity of a heterogeneous set of available aircraft. Given that each aircraft type has specific capacity, flying speed, and operating costs, this step determines for each flight leg of a (e.g., weekly) schedule which type of aircraft should operate it so as to maximize the total expected profit, that is, the difference between the expected passenger revenues and the operating costs. In certain cases, cargo revenues may also be considered for a passenger airline, but we assume here, for reasons of conciseness, that this is not the case. On the one hand, the operating costs (fuel costs, crew salaries, etc.) are, in general, well estimated and given by flight leg and aircraft type. On the other hand, the expected revenues depend on the passenger demand by pair of origin and destination, and the distribution of this demand over the itineraries offered by the airline. This distribution is determined by the itineraries desired by the passengers and the seating capacity of the aircraft assigned to each flight leg. The evaluation of this distribution is called the passenger flow evaluation. Observe that there is a cyclic interaction between the fleet assignment problem and passenger flow evaluation: the former problem requires revenues per leg to determine the capacity on each leg, while the latter requires a capacity on each leg to compute their revenues. This cyclic interaction has not been addressed directly in most works on the fleet assignment problem.

Hane et al. (1995) have introduced what is considered the basic fleet assignment model (FAM) for the daily fleet assignment problem, that is, when the same flight schedule replicates day after day. This model is a multi-commodity network flow model defined on a time−space network. The revenues are estimated by flight leg. The FAM is solved by a branch-and-bound heuristic that uses the dual simplex method with the steepest edge selection criterion, variable fixing, and node aggregation in the network. Various extensions of this model integrating certain features of the other steps of the operations planning process have been proposed. Clarke et al. (1996) incorporate certain aspects of aircraft maintenance and crew scheduling in the FAM. Desaulniers et al. (1997) and Rexing et al. (2000) consider time windows on the flight departure times which allow to slightly change the flight schedule. Ahuja et al.(2002) integrate the selection of multi-leg flights.

In practice, the flight schedules may differ from day to day, especially between weekdays and weekends. Furthermore, demand on certain flights may fluctuate significantly over the days of the week, justifying different assignments. For these reasons, some studies have considered the weekly fleet assignment problem. Barnhart et al. (1998) use a flight string model to solve the aircraft fleeting and routing problems simultaneously. Ioachim et al. (1999) consider the weekly fleet assignment problem with time windows and schedule synchronization constraints, where the flights with the same flight number (operating on different days) must have the same departure time. As an extension to this work, Bélanger et al. (2006a) handle revenues depending on the flight departure times and spacing constraints between the departure times of consecutive flights servicing the same market. Bélanger et al. (2006b) add homogeneity constraints that favor as much as possible the assignment of the same aircraft type to the flights with the same flight number. Smith and Johnson (2006) impose station purity, which limits the number of fleet types serving each station, to improve the robustness of the fleet assignment.

All FAMs cited above and earlier ones use an estimation of the revenues per flight leg even though numerous passenger itineraries contain two legs or more. This simplification neglects the interdependency between the flight legs: for instance, a passenger that must travel on a two-leg itinerary may be accepted on the first leg (together with the partial revenue it generates) but rejected on the second. It also approximates poorly the spill and recapture of the passengers. Spilling occurs when the number of passengers willing to take an itinerary exceeds the aircraft seating capacity of one of the leg in this itinerary. When the rejected passengers change their choice for another itinerary of the same airline, we talk about recapture. Clearly, there is a need to estimate the revenues by itinerary rather than by flight leg. Farkas (1996) and Barnhart et al. (2002) have developed the itinerary-based FAM (IFAM) which incorporates these network effects by adding decision variables on the itineraries that the passengers will choose. This system-optimization approach assumes that the airline can control the behavior of the passengers. Barnhart et al. (2009) have enhanced the IFAM by considering subnetworks and employing composite decision variables that represent the simultaneous assignment of fleet types to one or more flight legs of different subnetworks. Jacobs et al. (2008) incorporate origin and destination (O&D) network effects into the FAM by adding constraints to model the O&D revenue management process. The revenue approximation comes from a network flow model that maximizes the expected revenue subject to the seating capacity of each flight. This model also assumes that the airline controls the passenger behavior. The problem is solved using Benders decomposition. The linear relaxation of the FAM (master problem) is solved first and its solution is used in the network flow model (subproblem) to determine a new revenue approximation to add to the FAM. This process is repeated until the profit estimation from both models are close enough. The FAM is then solved to integrality and the network flow model is solved one more time to estimate the final expected profit. Sherali et al. (2006) provide a survey of the revenue modeling approaches for the fleet assignment problem.

The problem of treating the itineraries chosen by the passengers as decision variables arises from the fact that there exists a conflict between the airline's objective (profit maximization) and those of the passengers (cost minimization, schedule preferences, minimum number of legs, etc). The airline can use yield management tools to influence the passenger flow on its flights, but it cannot fully control the passengers. The fleet assignment problem should rather be viewed as a bilevel optimization problem incorporating the cyclic interaction mentioned above. At the first level, the airline assigns aircraft types to the flight legs which determines the operating costs. At the second level, the passengers choose their itineraries, allowing to compute the revenues.

To our knowledge, Dumas et al. (2009) were the first to propose a bilevel optimization approach that involves a user-optimization viewpoint to determine the revenues. Their iterative algorithm relies on a FAM with revenues per leg that are recomputed at each iteration using the passenger flow model (PFM) developed by Dumas and Soumis (2008). The PFM preserves the stochastic aspect of the demand and the temporal nature of the booking process, without controlling the passenger choice. The FAM is solved through a branch-and-bound heuristic similar to that of Hane et al. (1995) and the PFM is tackled by a fixed point method. The algorithm starts by solving the FAM with the traditional estimated revenues per leg. Given the resulting fleet assignment and the associated seating capacity for each leg, it then solves the PFM to reevaluate the expected revenues by itinerary. These revenues are then split by leg to redefine the FAM objective function and the FAM is reoptimized to begin a new iteration. For their computational experiments, Dumas et al. (2009) performed ten iterations of this algorithm to generate solutions (for a weekly fleet assignment problem involving more than 5,000 flight legs) yielding additional profits varying between 0.3 and 0.9 % of the total operating costs when compared to the profits obtained by solving the traditional FAM only once. The main drawback of this iterative solution process is the large increase in the computational times: on average, they are more than 10 times larger for the tests they realized on instances with around 5,000 flight legs. Their average computational time was about 2.5 h, which may seem quite reasonable for a planning problem. Note, however, that there exist much larger instances, involving more than 10,000 daily flights and requiring much larger computational times. Furthermore, as stated above, more complex variants of the fleet assignment problem (with time windows, maintenance requirements,…) are now tackled using mathematical programming techniques that can be combined with the passenger flow evaluation process. Solving large-scale instances of these variants is highly time-consuming compared to the basic fleet assignment problem. Finally, when designing a flight schedule for a whole season, the corresponding fleet assignment problem is typically solved to check schedule feasibility and to estimate expected profits. Given that schedule design is an iterative process that necessitates testing numerous scenarios (e.g., 100 scenarios over a one-month period), too large computational times for solving the fleet assignment problem with passenger flow evaluation limit the number of schedule scenarios that can be tested. For these reasons, we believe that the solution method of Dumas et al. (2009) has a limited applicability and should be improved.

In this paper, we revise the solution algorithm of Dumas et al. (2009) with the aim of reducing considerably the computational times while preserving solution quality. To do so, we propose to integrate the passenger flow evaluation within the branch-and-bound heuristic used to solve the FAM. The new algorithm consists of solving at each iteration a sequence of linear relaxations of the FAM before starting the branch-and-bound phase. After solving each linear relaxation in this sequence, variables are fixed, a restricted PFM is solved, and revenues are reevaluated. This integration allows a reduction of the number of iterations (and of the computational time) required to obtain solutions of quality comparable to that obtained by Dumas et al. (2009). The new algorithm is, thus, more suitable for solving very large-scale FAMs or fleet assignment problems involving complex features.

The paper is structured as follows. In Sect. 2, we summarize the solution algorithm of Dumas et al. (2009). In Sect. 3, we describe the proposed integrated algorithm. Computational results obtained on instances derived from a real-world dataset involving 5,180 legs and from a larger network involving more than 10,000 legs are reported in Sect. 4. Concluding remarks are presented in Sect. 5.

2 The solution method of Dumas et al. (2009)

The fleet assignment problem addressed in this paper, the same as in Dumas et al. (2009), integrates passenger flow evaluation to compute the passenger revenues and we refer to it as the fleet assignment problem with passenger flow evaluation. It can be stated as follows. Consider a flight network of an airline that spans a set S of stations (airports). Let L be the set of flight legs to operate over this network in a cyclic schedule spanning a time period (e.g., a day or a week). Here, cyclic means that the schedule repeats period after period and, thus, the computed solution must also be repeatable period after period. A leg in L is indexed by l or more explicitly by (o, d, t) to indicate that the leg is from origin station o to destination station d and departs at time t. To operate this schedule, aircraft of different types are available. Let F be the set of aircraft types and n_f the number of aircraft in fleet $f \in F$. Given the various characteristics (capacity, flying speed, autonomy, etc.) of the aircraft types, only a subset F_l of types can be assigned to a given leg $l \in L$. When fleet $f \in F_l$ is assigned to leg l, it incurs an operating cost of C_{fl}. The seats available in an aircraft are partitioned into fare classes for each leg. We denote by W the set of fare classes and by cap_{fl}^{w}, $w \in W$, $l \in L$ and $f \in F_l$, the number of seats of class w on leg l if operated by an aircraft of type f. The potential passengers are interested to buy tickets for a set I of itineraries, where an itinerary is defined by a sequence of legs in L and a fare class in W for each leg. For each itinerary $i \in I$, we know its demand d_i (number of passengers requesting it) and the average ticket price p_i paid by a passenger. To approximate the number of passengers spilled and recaptured on each itinerary, let q_{ij} be the proportion of passengers on itinerary $i \in I$ that, when rejected by a lack of capacity, are spilled onto itinerary $j \in I$. For a given $i \in I$, the sum $\sum_{j \in I} q_{ij}$ may be less than 100 % to model passengers that opt for another airline or another mode of transportation, that is, passengers that do not spill onto another itinerary in I. The fleet assignment problem with passenger flow evaluation consists of assigning an aircraft type $f \in F_l$ to each leg $l \in L$ such that the expected profits (expected revenues minus costs) are maximized, aircraft flow conservation by aircraft type is satisfied in the network at all times (that is, no deadhead flights can be added to balance the schedule), a minimum connection time is allowed between any pair of consecutive legs assigned to the same aircraft, and the number of available aircraft in each fleet is never exceeded. Computing properly the expected revenues involves determining the number of passengers that will buy a ticket on each itinerary $i \in I$, that is, evaluating the passenger flow in the network given the seating capacity of the aircraft assigned to each leg.

The ingredients of the solution algorithm proposed by Dumas et al. (2009) for solving the fleet assignment problem with passenger flow evaluation and the algorithm itself are discussed in the following paragraphs.

2.1 Fleet assignment model

The algorithm of Dumas et al. (2009) relies on a FAM that is an adaptation of the one introduced by Hane et al. (1995). The FAM of Hane et al. uses expected revenues per leg that depend on the aircraft type assigned to each leg. These revenues are approximated from a PFM solution by distributing the expected revenues for each itinerary $i \in I$ over the legs composing it and in proportion of their flight duration. Instead of considering expected revenues, Dumas et al. (2009) proposed to use expected revenue losses with respect to the maximum revenues that could be achieved by assigning a fictitious aircraft of infinite capacity. Let RL_{fl} be this expected revenue loss on leg $l \in L$ if an aircraft of type $f \in F$ is assigned to it. With these revenue losses, the objective function consists of minimizing the sum of the operational costs and the expected revenue losses. This is equivalent to maximizing the total profits.

The FAM is a multicommodity network flow model with side constraints that is defined over a time-space network. In this network (see Fig. 1), there exists a cyclic timeline for each aircraft fleet $f \in F$ and each station $s \in S$. A potential event corresponding to a departure or an arrival of an aircraft of type f at station s at time t is represented by a node $\{f, s, t\}$ on the timeline associated with station s and fleet f. To ensure the respect of the minimum connection time between two consecutive legs, the arrival time t includes this minimum time. It is denoted τ_{fodt} for a leg $(o, d, t) \in L$ assigned to an aircraft of type f. On a timeline, the nodes are ordered in chronological order. The time associated with the node preceding the time of an event occurring at time t is denoted by t^-, while that of the next node by t^+. Assuming that a timeline contains n nodes numbered from 1 to n, we let $t_n^+ = t_1$ and $t_1^- = t_n$. Let N be the set of nodes in this time-space network. As shown in Fig. 1, the network contains two types of arcs. Flight (diagonal) arcs represent the flight legs, each one linking the departure node of the corresponding leg on the departure station timeline to its arrival node on the arrival station timeline. Ground (horizontal) arcs represent idle periods at stations and link consecutive event nodes on every station timeline (as well as the last event node to the first event node at a station).

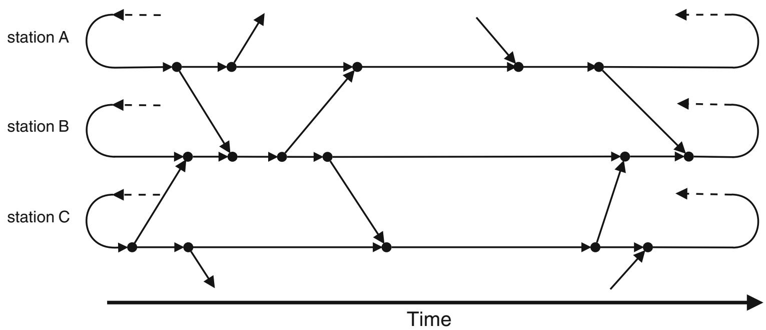

Fig. 1 Example of a time-space network with three stations

To impose aircraft availability per fleet, the aircraft of each fleet used in the solution needs to be counted once at a specific time $\bar{t}$. Let $[t_{sf}, t^{+}_{sf}]$, $s \in S$ and $f \in F$, be the time interval that is delimited by two consecutive events on the timeline associated with station s and fleet f, and such that $\bar{t} \in [t_{sf}, t^{+}_{sf}]$. Furthermore, let O_f be the set of flight legs that can be operated by an aircraft of type f and whose operating time (including the subsequent minimum connection time) spans $\bar{t}$.

The proposed FAM involves two types of decision variables (one type for the flight arcs and the other for the ground arcs). For each leg $l = (o, d, t) \in L$ and each fleet $f \in F_l$, there is a binary variable X_{fl} (or X_{fodt}) that takes value 1 if fleet f is assigned to leg l and 0 otherwise. For each fleet $f \in F$, each station $s \in S$ and each time interval $[t, t^{+}]$ on the timeline associated with f and s, there is a nonnegative variable $Y_{fstt^{+}}$ indicating the number of aircraft of type f on the ground at station s in this interval. With this notation, the FAM is as follows:

$$\min \sum_{l \in L} \sum_{f \in F_l} X_{fl}(C_{fl} + \mathrm{RL}_{fl}) \tag{2.1}$$

$$\text{subject to:} \sum_{f \in F_l} X_{fl} = 1, \quad \forall l \in L, \tag{2.2}$$

$$\sum_{o \in S} \sum_{t' : \tau_{fost'} = t} X_{fost'} + Y_{fst^{-}t} - \sum_{d \in S} X_{fsdt} - Y_{fstt^{+}} = 0, \quad \forall \{f, s, t\} \in N, \tag{2.3}$$

$$\sum_{l \in O_f} X_{fl} + \sum_{s \in S} Y_{fst_{sf}t^{+}_{sf}} \le n_f, \quad \forall f \in F, \tag{2.4}$$

$$X_{fl} \in \{0, 1\}, \quad \forall l \in L, f \in F_l, \tag{2.5}$$

$$Y_{fstt^{+}} \ge 0, \quad \forall \{f, s, t\} \in N. \tag{2.6}$$

The objective function (2.1) aims at minimizing the sum of the operating costs and the expected revenue losses. Constraints (2.2) ensure the assignment of exactly one fleet to each flight leg in L. Constraints (2.3) enforce flow conservation in the network for each aircraft type. Aircraft availability by fleet is imposed through constraints (2.4). Binary and nonnegativity requirements (2.5)–(2.6) restrict the feasible domains of the variables. Note that constraints (2.3) and (2.5) imply the integrality of the $Y_{fstt^{+}}$ variables.

The FAM (2.1)–(2.6) can be solved by a branch-and-bound algorithm. To limit the computational times for large-scale instances, variable fixing is often applied at the beginning of the solution process. The details of this algorithm are provided in Sect. 3.

2.2 Passenger flow model

Dumas and Soumis (2008) developed a PFM that allows the evaluation of the expected revenues yielded by a FAM solution. It aims at determining the expected number of passengers on each itinerary $i \in I$ taking into account the seating capacity offered by the FAM solution on each flight leg $l \in L$ and for each fare class $w \in W$. The PFM requires the following inputs:

1. The flight schedule L and the aircraft type assigned to each flight leg $l \in L$;
2. The seating capacity in each fare class $w \in W$ on each leg $l \in L$ (a pair (w, l) of fare class and leg is called an arc by Dumas and Soumis 2008);
3. The distribution of the passenger demand for each itinerary $i \in I$ which is seen as a random variable (following, e.g., a normal distribution);
4. The temporal distribution of the booking requests for each itinerary $i \in I$;
5. The proportion q_{ij} of passengers spilled from a closed itinerary $i \in I$ to another itinerary $j \in I$.

An arc is said to be closed when the number of passengers booked on this arc reaches its capacity and an itinerary is closed when at least one of its arcs is closed. Notice that, in the PFM of Dumas and Soumis (2008), the passengers' objectives (cost minimization, schedule preferences, minimum number of legs, etc.) are not explicitly modeled. Instead, the distribution of the overall demand of a given market on all its possible itineraries is assumed to be known and is implicitly taken into account in the passenger demand distribution of each itinerary. The PFM is partitioned into disjoint time slices to preserve the temporal aspect of the booking process. In each time slice, it is composed of a system of nonlinear equations that involves several variables. There are variables indicating the expected number of booking requests in that time slice for each itinerary and for each arc, as well as variables providing the probability that an itinerary or an arc is closed during that time slice. The system contains two main sets of equations. The first set expresses, for each itinerary, the expected number of booking requests in the current time slice as a function of its demand, the temporal distribution of its booking requests, the total number of booking requests on the other itineraries, the probability that each itinerary is closed in that time slice, and the proportions of the spilled passengers. The second set of equations relate the probability that an itinerary is closed in the current time slice to the probability that one of its arcs is closed. The other equations allow the computation of various quantities used in the first two sets of equations (for more details, see Dumas and Soumis 2008). The equation system is solved using a fixed point method (see, e.g., Burden and Faires 2011, chapter 10) and its solution provides the passenger flow in the network at the end of the time slice.

The solution process of the whole PFM proceeds sequentially, solving the PFM restricted to the time slices in chronological order. In each time slice, the fixed point method starts from an initial solution. In this solution, the total number of booking requests for an itinerary is set to the itinerary demand for that time slice, whereas the probability that an itinerary or an arc is closed is set to its final value in the previous time slice (or to 0 if it is the first slice). The values of the variables are then updated one after the other using the system of equations until a stopping criterion is met. Computational experiments performed by Dumas and Soumis (2008) over different networks showed that 15 iterations is sufficient to provide an acceptable solution. The expected revenues yielded by the FAM solution are derived from the passenger flow at the end of the last time slice.

The PFM solution allows to compute the expected revenues per leg as a function of the aircraft type (seating capacity) assigned to each leg. This information is incomplete for the FAM that requires the expected revenues (or rather the expected revenue losses RL_{fl}) for each leg $l \in L$ and each aircraft type $f \in F_l$ that can be assigned to it. To obtain all these revenue losses, one can solve a large number of PFM instances, namely, one for each valid combination of leg and aircraft type. This approach would, however, yield very large computational times. To speed up these computations, Dumas et al. (2009) introduced a local version of the PFM. On the tested instances, they observed that this version can achieve a speedup factor of 400 without losing too much solution accuracy.

The local version of the PFM is defined for a pair of leg $l \in L$ and aircraft type $f \in F$. It is similar to the full PFM but restricted to a subnetwork denoted (A_l, I_l). The set I_l contains all itineraries involving leg l as well as all itineraries that can recover passengers from these first itineraries. The set A_l is composed of all arcs used by at least one itinerary in I_l. Considering only the itineraries in I_l is not sufficient to determine the passenger flow on the arcs of A_l. Indeed, there are other itineraries that involve these arcs or whose demand can spill onto them. Consequently, to take these other itineraries into account, the full PFM is first solved (using the fleet assignment of the FAM solution) and, after each time slice, the following information are kept in memory: for each itinerary $i \in I$, the expected number of passengers accepted and rejected in the time slice and, for each arc, the probability that it is closed. When solving the local PFMs thereafter, these stored information are used to preassign passengers to their corresponding itineraries.

The PFM can be defined and solved even if the FAM solution is fractional. Indeed, the FAM solution provides seating capacity per fare class and flight leg. When a FAM solution assigns a convex combination of aircraft types to a leg, its seating capacity per fare class can be computed as the corresponding convex combination of the capacities of these aircraft types. We exploit this possibility in the method proposed for solving the fleet assignment problem with passenger flow evaluation.

2.3 Fleet assignment with external PFM reevaluation

The iterative procedure used by Dumas et al. (2009) for solving the FAM with an external reevaluation of the revenues by the PFM is summarized in Fig. 2. More precisely, a first evaluation of the revenues per leg is performed to compute the initial vector of expected revenue losses RL^0. For a leg $l \in L$ and an aircraft type $f \in F_l$, the component RL^+_{sf} of this vector is computed as the expectation of the demand that exceeds the capacity of an aircraft of type f multiplied by a weighted average of the ticket price paid by the passengers spilled from the itineraries containing leg l. The iterative procedure starts at iteration 0 that serves as the reference point to compare the results. At iteration $k \geq 0$, the following steps are performed:

- Using the vector RL^k in its objective function (2.1), the FAM (2.1)–(2.6) is solved by a commercial mixed integer programming solver (CPLEX in our case). The solution process can be stopped prematurely whenever a feasible solution is found and its value is close enough to the current lower bound on the

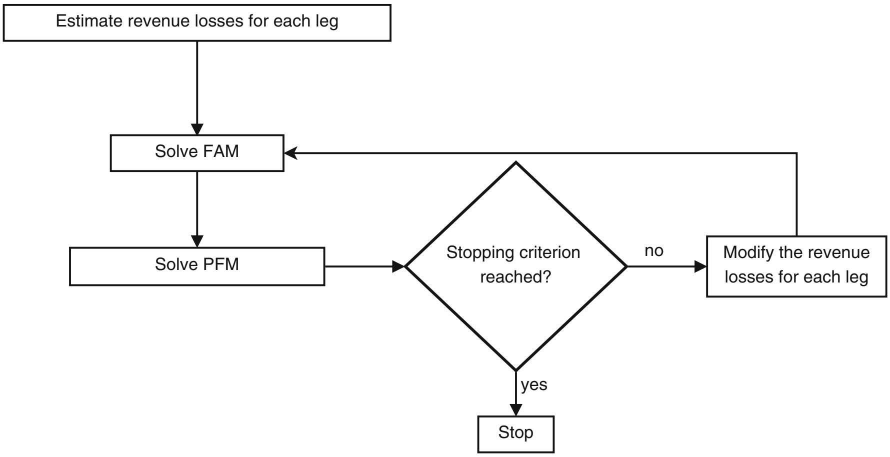

Fig. 2 The method of Dumas et al. (2009): fleet assignment with external PFM reevaluation

optimal value, that is, if it respects a tolerance on the optimality gap. The computed fleet assignment solution is denoted X^k.

- The full PFM is executed for the solution X^k and the numbers of booking requests and of spilled passengers for each itinerary and each time slice are kept in memory. This step allows to compute the expected revenues of the current solution X^k and, therefore, the value of this solution.
- The stopping criterion is then checked. In Dumas et al. (2009), the iterative process stops whenever a maximum number of iterations is reached.
- When the process is not halted, it continues by reevaluating the revenue losses per leg according to the current fleet assignment solution X^k. To do so, the local PFM is first executed for each leg $l \in L$ and each fleet $f \in F_l$. In this execution, the numbers of booking requests and of spilled passengers per time slice for each itinerary that is not part of the subnetwork of leg l (information stored in memory from the last solution of the full PFM) are fixed as constant. The revenue loss vector obtained from these computations is denoted $\overline{\mathrm{RL}}^{k+1}$. Rather than using directly this vector to redefine the objective function of FAM, Dumas et al. (2009) use a convex combination of this vector and the previous one RL^k: the new vector is given by $\mathrm{RL}^{k+1} = \alpha\overline{\mathrm{RL}}^{k+1} + (1-\alpha)\mathrm{RL}^k$, where $\alpha \in [0, 1]$ is a parameter. This smoothing strategy allows to avoid too large variations of the components of the revenue loss vector from one iteration to the next. Numerical experiments realized by Dumas et al. (2009) show that the value $\alpha = 0.3$ provides fast convergence and good quality solutions.

Notice that the method of Dumas et al. (2009) does not guarantee a decrease of the FAM optimal (total revenue loss) value from one iteration to the next. Nevertheless, its smoothing strategy helps, in practice, generating a sequence of decreasing values until getting close to optimality.

3 Fleet assignment with internal PFM reevaluation

In this section, we present our proposed method for solving the fleet assignment problem with passenger flow evaluation and highlight how it differs from the method of Dumas et al. (2009). Our goal is to produce similar quality solutions as those obtained by Dumas et al. (2009) but in faster computational times.

For solving large-scale FAMs, variable fixing is commonly used as a heuristic to reduce computational times (Hane et al. 1995; Sherali et al. 2009). The general idea of the proposed method is to fix variables and reevaluate revenues with the PFM simultaneously. In this way, the decisions made after each revenue reevaluation are based on updated revenues and should, therefore, yield a better solution than the solution obtained without any revenue reevaluation. Figure 3 summarizes this method with internal PFM reevaluation. Instead of solving directly the FAM (2.1)–(2.6) with integrality requirements like in the method of Dumas et al. (2009), we begin by solving only its linear relaxation. Then fleet assignment variables are fixed and the revenue losses per leg are reevaluated for the legs impacted by the imposed decisions. The modified linear relaxation is solved again. This process repeats until one of the three following conditions hold: a predefined number of iterations is reached, a predefined number of variables is fixed, or no more variables can be fixed. The reduced FAM resulting from this variable fixing process and the PFM reevaluations is then solved using a branch-and-bound algorithm. Below, we give details on the variable fixing strategy (Sect. 3.1), the revenue losses update (Sect. 3.2), and the branch-and-bound algorithm (Sect. 3.3).

3.1 Variable fixing strategy

In a typical solution of the linear relaxation of the FAM (2.1)–(2.6), there is a variable X_{fl} equal to 1 for around 80 % of the legs $l \in L$, that is, the aircraft type is not fixed for around 20 % of the legs. As in Sherali et al. (2009), one can decide to leave unfixed the variables equal to 1 to offer more leeway to the MIP solver.

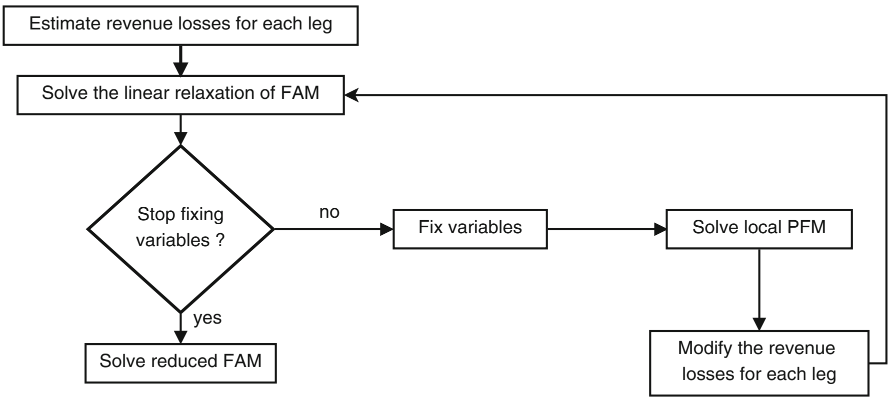

Fig. 3 The proposed method: fleet assignment with internal PFM reevaluation

However, because these variables represent a large proportion of the positive-valued variables and most of them remain at 1 when only fractional-valued variables are fixed, we have chosen to fix some of them at each iteration in order to substantially reduce the size of the FAM. The number of variables to fix at each iteration is limited by a parameter denoted V_1. Furthermore, we also impose that at most a given percentage PV_1 of the variables equal to 1 for a given aircraft type $f \in F$ be fixed to 1 (this parameter was fixed to 75 % for our tests). This limit per fleet ensures a relatively good distribution of the fixed variables among the different fleets. To change the solution from one iteration to the next, we also fix a certain number of fractional-valued variables X_{fl}. These variables are first sorted in decreasing order of their value, and then the first V_2 of them are fixed to 1 as long as their value is greater than or equal to a predefined minimum threshold B_{inf}. No other variables are fixed, which means that no variables are fixed at 0. After fixing to 1 the chosen variables, the linear relaxation of the FAM is solved to obtain a new linear relaxation solution that may still contain fractional-valued variables. In this case, variable fixing is repeated. We allow $V_{\max}$ variables to be fixed overall. When this limit is reached, the reduced FAM is solved with integrality requirements. Notice that $V_{\max}$ should be large enough to obtain relatively fast computational times but not too large to ensure that variable fixing does not yield an infeasible reduced FAM. In Sect. 4.2, we discuss the values attributed to the parameters V_1, V_2, $V_{\max}$, and $B_{\inf}$ for our tests.

3.2 Revenue losses update

After solving the linear relaxation of FAM and fixing certain variables, the revenue losses per leg are reevaluated using the local version of the PFM. The method described in Sect. 2.2 is then used to update the vector RL in the objective function of the FAM.

Two observations allow to accelerate this process. Firstly, the revenue losses of the legs fixed to 1 do not have any impact on the optimization process. They are now fixed and considered as constant in the objective function. Hence, there is no need to reevaluate them until the end of the solution process, where the revenues for each leg will be computed once again with the full PFM to obtain the value of the final solution. Secondly, significant changes to the revenue losses do not occur for all legs after fixing certain variables. To limit the number of legs for which revenue losses need to be reevaluated, we consider only the legs that can be directly impacted by the fixed variables. For a leg $l \in L$ for which a variable X_{fl} was fixed at this iteration, these legs are those in I_l (that is, the set of itineraries containing l and those that can recover passengers from these itineraries).

3.3 Branch-and-bound algorithm and iterative process

Once a sufficient number of variables are fixed, the reduced problem is solved by a commercial MIP solver, namely, CPLEX. The search tree is first explored using a depth-first search strategy until finding a first integer solution. If the value of this solution respects a tolerance on the optimality gap, then the solution process is terminated. Otherwise, the exploration of the search tree is pursued using a combination of depth-first search and best-first search strategies until reaching a predefined maximum number of nodes (set to 500 for our tests). More precisely, we set the tolerance backtracking parameter of CPLEX (CPX_PARAM_BTTOL) to 0.5. This allows to explore sufficiently each branch using a depth-first search as long as the branch remains promising and to switch to a more promising branch as soon as the current branch does not seem promising anymore.

When a feasible solution respecting the tolerance optimality gap is found, the full PFM derived from the fleet assignment solution is solved a final time to evaluate the expected revenues on all flight legs and the total expected profits of the solution. This final evaluation allows a fair comparison between the solutions produced by our method and those of Dumas et al. (2009).

To improve the fleet assignment solution, the method of Dumas et al. (2009) with external PFM reevaluation performs multiple iterations in which the expected revenue losses per leg are reevaluated at each iteration. In the proposed method with internal PFM reevaluation, these revenue losses are reevaluated less than once for a large number of legs. Thus, we propose to also embed our method into an iterative process that will repeat several times the algorithm described in Fig. 3, using as the starting point of each iteration, the fleet assignment solution computed in the previous iteration. As it will be shown by the computational results presented in the next section, this iterative process converges very rapidly and, therefore, allows to preserve the quality of the solutions produced by the Dumas et al. (2009) method while reducing the overall computational times.

4 Computational results

This section reports the results of various computational experiments that we conducted to assess the performance of the proposed method with internal PFM reevaluation against that of the method of Dumas et al. (2009) with external PFM reevaluation. In this section, we refer to these methods as the internal and the external method, respectively. Both methods were implemented in C++ and rely on the commercial MIP solver CPLEX, version 12.4. All tests were performed using a single core of an Intel Xeon X5670 processor clocked at 2.93 GHz and 24 Gb of RAM.

4.1 Instances

For our computational experiments, we use a part of Air Canada's flight network in 2002. It contains 5,180 legs operated by 205 aircraft of 15 different types. The passengers were distributed among 23,948 itineraries and 3 fare classes. As this is the unique large-scale network for which we have all necessary information to solve the PFM, we applied the demand perturbations suggested by Dumas et al. (2009) to create two other sets of expected demand on each itinerary. These perturbations are obtained by multiplying the expected demand for each itinerary by a random number taken from a uniform distribution in the interval [0.55, 1.55]. The new demands are then scaled so that the overall modified demand be equal to the total original demand. In 2002, demand was relatively low with an average load factor of around 74.8 % for our network. In comparison, Air Canada had an average load factor of 82.7 % in 2012 (Air Canada 2012, p.2), which is closer to their usual one. Consequently, to create demands yielding higher load factors, we introduce a parameter that multiplies the expected demand for each itinerary. For each of the three demand structures, we use five different parameter values, resulting in a total of 15 instances.

To show that the proposed method can yield faster computational times than the method of Dumas et al. (2009) when solving large instances, we also created a larger network by duplicating each leg and each aircraft of the 5,180-leg network mentioned above. The departure time of each new leg was shifted forward by 73 minutes to avoid duplicated legs at the same time. Itineraries were also copied and new ones were created to take into account the new possibilities. Finally, we adjusted the proportion q_{ij} of passengers spilled from a closed itinerary $i \in I$ to another itinerary $j \in I$, keeping the same overall recapture rate for a given itinerary. This new network contains 10,360 legs, 410 aircraft of 15 types, and 65,169 itineraries. From this network, we derived 9 instances with different demand structures as described above.

4.2 Parameter values

As mentioned in Sect. 3.1, the variable fixing strategy of the internal method relies on several parameters whose values need to be calibrated. To reduce the computational times, we integrate the same variable fixing strategy into the external method. However, as this method does not modify the revenue losses per leg during the FAM solution process, the best parameter setting may differ for each method. Consequently, we calibrated the parameter values independently for each method and also for each network. The selected values are given in Table 1. To assess the impact of the parameter values on the quality of the solutions produced by the internal method and its computational time, see Sect. 4.4 that reports the results of a sensitivity analysis on these parameters for the 5,180-leg network.

4.3 Comparative results for the 5,180-leg network

For each pair of demand structure and load factor ($3 \times 5 = 15$ pairs overall) for the 5,180-leg network, we solved the problem instance using both internal and external solution methods. To measure the quality of a computed solution, we compare its expected profit with that of the solution produced by solving only once the fleet assignment problem without reevaluating the expected revenues per leg, that is, solution X^0 of the external method. As Air Canada (like most airlines) was struggling for profitability in 2002, it would not be representative to provide profit improvement in relative value (profits were negative and close to 0). We rather report the profit improvement in percentage of the costs of the initial solution X^0.

For each method, Fig. 4 illustrates the average of the profit improvement for the solutions computed throughout the solution process (each point corresponds to the

Table 1 Parameter values for both internal and external methods

Parameter	Description	5,180-leg network		10,360-leg network	
		Value for internal	Value for external	Value for internal	Value for external
$V_{\max}$	Max. number of variables to fix overall	3,000	3,500	7,000	7,000
V_1	Max. number of variables equal to 1 fixed at once	350	300	600	500
V_2	Max. number of fractional-valued variables fixed at once	200	300	400	350
$B_{\inf}$	Min. value threshold for fixing a variable	0.8	0.75	0.75	0.75

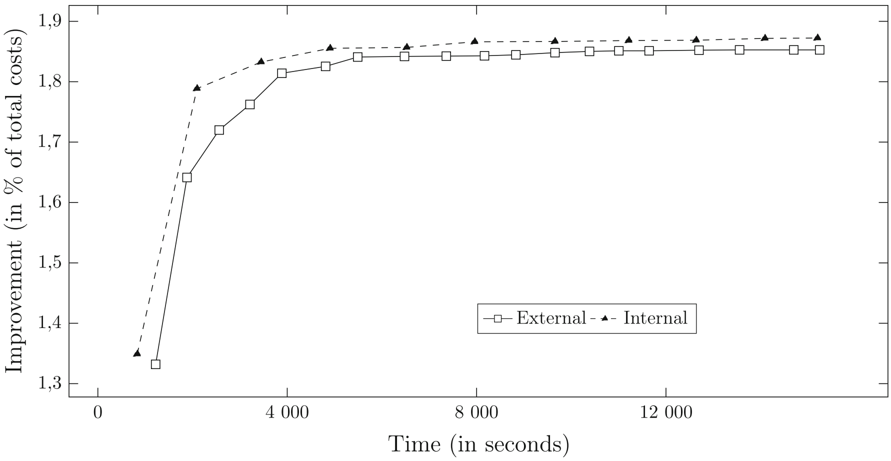

Fig. 4 Average results of the internal and external methods (5,180-leg network)

best solution obtained up to a given iteration). In this figure, the horizontal axis indicates the average computational time at the end of each iteration. These two curves clearly show that the internal method converges more rapidly than the external method and to a higher improvement value. It thus requires fewer iterations to obtain solutions of similar quality, yielding shorter computational times.

In Table 2, we present the results (profit improvement in percentage of the total costs of the initial solution X^0 and computational time in seconds) obtained for each instance after **17** iterations of the external method (no improvements were observed in the few next iterations) and those obtained after 4 and 11 iterations of the internal method. The last line of this table reports averages over all instances. After four iterations of the internal method, we obtain a solution that is slightly better on average than the best one of the external method, and this in just over a third of the

Table 2 Detailed results of the internal and external methods

Demand structure	Load multiplier	Average load factor (%)	External method		Internal method			
			17 iterations		4 iterations		11 iterations	
			Imp. (%)	Time (s)	Imp. (%)	Time (s)	Imp. (%)	Time (s)
I	1.05	76.5	1.26	9,550	1.23	3,315	1.26	10,687
	1.1	79.0	1.33	9,521	1.35	3,429	1.35	9,383
	1.15	81.1	1.58	10,488	1.53	3,847	1.58	11,306
	1.2	82.9	1.72	10,120	1.74	4,985	1.74	20,793
	1.25	84.5	1.89	20,318	1.93	5,092	1.93	17,005
II	1.05	77.6	1.64	9,102	1.62	3,986	1.62	10,659
	1.1	79.8	1.72	11,249	1.70	4,137	1.71	13,438
	1.15	81.6	1.89	11,036	1.85	4,557	1.91	15,560
	1.2	83.1	2.18	19,730	2.20	8,054	2.20	20,824
	1.25	84.4	2.51	22,144	2.51	7,017	2.54	17,752
III	1.05	77.5	1.53	10,112	1.51	4,095	1.55	12,070
	1.1	79.7	1.81	10,590	1.81	4,886	1.81	14,125
	1.15	81.5	2.02	14,203	2.04	5,473	2.04	15,024
	1.2	83.1	2.20	10,397	2.25	4,141	2.26	12,639
	1.25	84.3	2.50	24,770	2.54	6,558	2.57	26,745
Average			1.853	13,555	1.855	4,905	1.872	15,201

Table 3 Statistics on the FAM solution process (5,180-leg network)

	External method	Internal method
Linear relaxation time per iteration (s)	95.1	96.2
Total time per iteration (s)	783	912
No. branch-and-bound nodes (reduced MIP)	22.0	23.8
No. of cuts (reduced MIP)	13.4	11.9
No. of PFM reevaluations	–	6.6

time. After 17 iterations, the external method achieves an average profit improvement of 2,390,667$ compared to 2,394,000$ for the internal method with four iterations. While their average profit improvements are very close, the internal method requires 63.8 % less time than the external method in this setting. For this network, the internal method reaches after 15,201 seconds an average profit improvement that is slightly larger than that of the external method. Note that, with both methods, the average improvement is substantial compared to the traditional approach for solving the fleet assignment problem without reevaluating the revenues per leg. We also observe that the improvement increases with the load factor, that is, when spill and recapture are very active. In this case, a first approximation of the expected revenues per leg is often not very good.

Even if both methods alternate between solving the FAM and the PFM, they do not spend the same proportion of time on each problem component. The external method solves the full PFM only once per iteration and spends on average 88.1 % of the total time on the FAM, devoting only 11.9 % to the PFM. On the other hand, the internal method solves the full PFM once per iteration but also the local version of the PFM for a subset of the legs each time that variables are fixed, that is, about 6–8 times per iteration. A larger proportion of the total time is, therefore, necessary to solve the PFM, namely, about 33.7 %.

To conclude this section, Table 3 reports some statistics on the solution process of the FAM within the tested methods. For each method, we give the time in seconds required to solve the linear relaxation, the total computational time per iteration in seconds (excluding PFM), the total number of branch-and-bound nodes explored and the total number of cuts generated by CPLEX while solving the reduced MIP, and, finally, the number of PFM reevaluations (only for the internal method). All these results correspond to averages per iteration and instance. From these results, we observe that the FAM solution process behaves very similarly in both methods except that the internal method performed an average of 6.6 PFM reevaluations while the external method did not use such reevaluations. In particular, note that the average number of branch-and-bound nodes per iteration is relatively low (22.0 and 23.8 for the internal and the external method, respectively). However, as reported in Dumas et al. (2009), there are very few branch-and-bound nodes explored in most iterations (10 or less) while some of them require a relatively large number. In the former cases, the first integer solution found is within the tolerance on the optimality gap while in the latter cases, it exceeds this tolerance.

4.4 Sensitivity analysis

To evaluate the impact of the parameter values used in the internal method on the solution quality and the computational time, we performed a sensitivity analysis on these values considering 9 instances of the 5,180-leg network (those corresponding to load multipliers 1.05, 1.15, and 1.25). Starting from the values used for the external method ($V_{\max} = 3{,}500$, $V_1 = 300$, $V_2 = 300$, $B_{\inf} = 0.75$), we varied the value of a single parameter at a time and solved the instances with the internal method allowing four iterations. The average profit improvements and the computational times obtained are reported in Table 4. In this table, the first row provides the results for the reference values of the parameters. Then, each of the following four block of rows reports the results for different values (in bold) of a single parameter while the others stay at their reference value. When comparing the results in a block, the first row should also be considered. Finally, the last row gives the results for the best value of each parameter, that is, the results reported in the previous section.

Let us discuss the results for each parameter. Decreasing the maximum number of variables fixed overall, $V_{\max}$, does not change much the solution quality. At each iteration, fixing fewer variables yields a larger reduced MIP and, thus, the solution process is less heuristic. On the other hand, the revenue losses are revised less frequently, loosing precision on the coefficients of the objective function. The average computational time seems to vary arbitrarily because fixing fewer variables reduces the time spent in the variable fixing phase (including the time for the PFM reevaluations) but increases the time required to solve the reduced MIP that contains more variables. One would expect that the latter time would increase much more rapidly than the former time, but the first solution found by the MIP solver often meets the optimality gap tolerance.

We remark that decreasing the value of V_2, the maximum number of fractional-valued variables fixed at once, the solution quality improves at the expense of longer computational times. Indeed, the solution quality improves because the average value of the variables fixed increases (the decisions are thus safer) and the PFM is reevaluated more often. The same behavior is not observed when decreasing the value of V_1 because this parameter controls the variables equal to 1.

The minimum value threshold $B_{\inf}$ is closely related to parameter V_2 because both limit the number of fractional-valued variables that can be fixed at once. We remark that setting $B_{\inf} = 0.8$ or 0.85 yields a slightly higher average solution quality. At the opposite, a lower value for $B_{\inf}$ increases the risk of imposing bad decisions and producing solutions of poor quality. We observe that the average computational time varies arbitrarily.

Finally, observe that the selected parameter values (last row in Table 4) yield the best average profit improvement over all parameter combinations and an average computational time that is more or less in the middle of the times obtained with the other configurations.

Table 4 Sensitivity analysis on the parameter values for the internal method

Parameters				Average improvement (% of costs)	Average time (s)
V_{max}	V_1	V_2	B_{inf}		
3,500	300	300	0.75	1.819	5,089
2,500	300	300	0.75	1.815	6,312
3,000	300	300	0.75	1.819	4,065
4,000	300	300	0.75	1.817	5,551
3,500	**250**	300	0.75	1.830	4,911
3,500	**350**	300	0.75	1.833	4,905
3,500	**400**	300	0.75	1.826	4,144
3,500	300	**100**	0.75	1.836	6,031
3,500	300	**200**	0.75	1.829	5,007
3,500	300	**400**	0.75	1.803	3,992
3,500	300	300	**0.7**	1.820	4,478
3,500	300	300	**0.8**	1.824	4,495
3,500	300	300	**0.85**	1.823	5,157
3,000	**350**	**200**	**0.8**	**1.862**	**4,882**

4.5 Comparative results for the 10,360-leg network

To assess if the computational time reductions yielded by the internal method and observed for the 5,180-leg network can be reproduced for larger instances, we created a network with 10,360 flight legs (as described in Sect. 4.2) and three demand structures for this network. For each demand structure, three different load multipliers (1.05, 1.15, 1.25) were applied to generate a total of 9 instances. These instances were then solved using the external and the internal method. Figure 5 presents graphically the average profit improvement in percentage of the costs of the initial solution obtained at each iteration by each method. As for the 5,180-leg instances, the internal method converges much more rapidly than the external method and requires fewer iterations to obtain solutions of similar quality. For this network, both methods reaches a similar profit improvement after 70,000 s (no further improvements occur in the next iterations).

Detailed results for all instances are reported in Table 5. On average, the internal method computes in 4 iterations a solution of similar quality to the best solution computed by the external method in 11 iterations, saving 54.7 % of the computational time. These solutions correspond to an average profit improvement of around 7,900,000$. Here again, we observe that the profit improvement and the computational time reduction (derived after four iterations of the internal method) increase with the average load factor.

Doubling the size of the network has a bigger impact on the computational time devoted to the PFM than on that devoted to the FAM. Indeed, the FAM time is partly controlled by the size of the reduced MIP which was kept relatively low for the large network. Consequently, the proportion of time devoted to the PFM

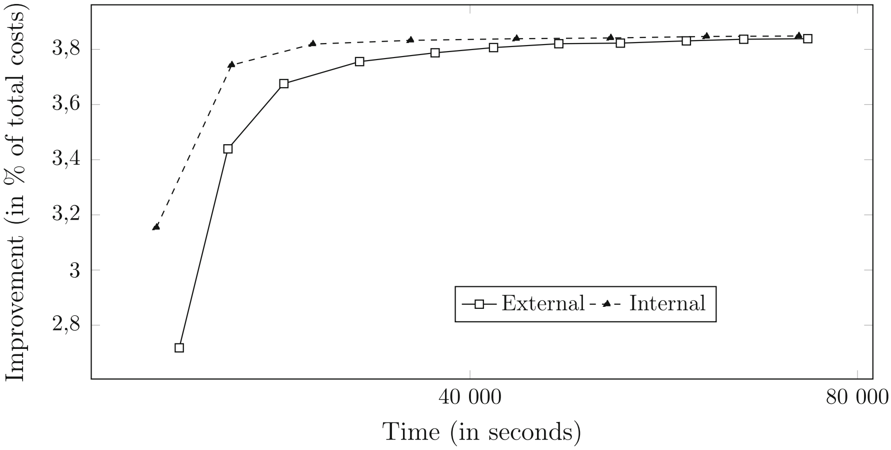

Fig. 5 Average results of the internal and external methods (10,360-leg network)

increased in both methods. For the 10,360-leg network, the external method spent 16.5 % of the total time solving the PFM while it required 11.9 % for the 5,180-leg network. The increase is much sharper for the internal method that performed on average twice the number of PFM reevaluations in each iteration. As a result, solving the PFM in the internal method required 61.6 % of the total time for the large network while it took only 33.7 % for the 5,180-leg network.

The results of the experiments conducted on both networks show that the internal method can yield substantial average time reductions compared to the external method (63.8 and 54.7 % for the networks with 5,180 and 10,360 legs, respectively). We believe that the internal method would also produce similar time savings for larger instances or for fleet assignment problems involving more complex features that would require larger computational times.

5 Conclusion

In this paper, we addressed the fleet assignment problem with passenger flow evaluation, an airline fleet assignment problem that computes revenues using a PFM. Recently, Dumas et al. (2009) developed an iterative solution method for this problem that solves alternately a FAM with expected revenues per flight leg and a PFM to revise the revenues per leg. It appears, however, to be too time-consuming for solving very large-scale instances or instances of problem variants involving additional features such as time windows or aircraft maintenance requirements. Our goal was, thus, to develop an alternative method that reevaluates the revenues while solving the FAM in the hope of reducing substantially the computational times. Our method with internal revenue reevaluations has turned out to be efficient. It converges much more rapidly to solutions of the same quality as those produced by the method of Dumas et al. (2009), yielding much faster computational times (around 2–3 times faster).

Table 5 Detailed results of the internal and external methods (10,360-leg network)

Demand structure	Load multiplier	Average load factor (%)	External method		Internal method			
			11 iterations		4 iterations		8 iterations	
			Imp. (%)	Time (s)	Imp. (%)	Time(s)	Imp. (%)	Time (s)
I	1.05	77.0	2.41	49,682	2.30	24,856	2.31	54,430
	1.15	81.1	3.22	54,346	3.17	30,692	3.21	65,821
	1.25	84.1	4.59	79,106	4.60	36,785	4.61	82,227
II	1.05	77.8	2.67	73,943	2.64	28,916	2.70	59,358
	1.15	81.4	4.01	76,199	4.08	44,451	4.08	82,344
	1.25	84.3	5.26	114,190	5.22	34,897	5.23	82,114
III	1.05	77.8	2.83	59,986	2.87	32,137	2.88	65,341
	1.15	81.3	4.03	76,284	4.05	35,816	4.05	94,751
	1.25	84.1	5.52	90,056	5.57	36,280	5.57	78,981
Average			3.838	74,866	3.833	33,870	3.849	73,930

References

Ahuja R, Goodstein KJ, Mukherjee A, Orlin JB, Sharma D (2002) A very large-scale neighborhood search algorithm for the combined through-fleet-assignment model. INFORMS J Comput 19(3): 416–428

Air Canada (2012) Annual Report 2012. http://www.aircanada.com/en/about/investor/documents/2012_ar.pdf

Barnhart C, Boland NL, Clarke LW, Johnson EL, Nemhauser GL, Shenoi RG (1998) Flight string models for aircraft fleeting and routing. Transp Sci 32(3):208–220

Barnhart C, Farahat A, Lohatepanont M (2009) Airline fleet assignment with enhanced revenue. Oper Res 57(1): 231–244

Barnhart C, Kniker T, Lohatepanont M (2002) Itinerary-based airline fleet assignment. Transp Sci 36(2):199–217

Bélanger N, Desaulniers G, Soumis F, Desrosiers J (2006a) Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues. Eur J Oper Res 175(3):1754–1766

Bélanger N, Desaulniers G, Soumis F, Desrosiers J, Lavigne J (2006b) Weekly airline fleet assignment with homogeneity. Transp Res Part B: Methodol 40(4):306–318

Burden RL, Faires JD (2011) Numerical analysis, 9th edn. Brooks/Cole Cengage Learning, Boston

Clarke L, Hane C, Johnson E, Nemhauser G (1996) Maintenance and crew considerations in fleet assignment. Transp Sci 30(3):249–260

Desaulniers G, Desrosiers J, Dumas Y, Solomon MM, Soumis F (1997) Daily aircraft routing and scheduling. Manag Sci 43(6):841–855

Dumas J, Aithnard F, Soumis F (2009) Improving the objective function of the fleet assignment problem. Transp Res Part B 43(4):466–475

Dumas J, Soumis F (2008) Passenger flow model for airline networks. Transp Sci 42(2):197–207

Farkas A (1996) The influence of network effects and yield management on airline fleet assignment decisions. Doctoral thesis, Massachusetts Institute of Technology, Cambridge

Hane CA, Barnhart C, Johnson EL, Marsten RE, Nemhauser GL, Sigismondi G (1995) The fleet assignment problem: solving a large-scale integer program. Math Program 70(2):211–232

Ioachim I, Desrosiers J, Soumis F, Bélanger N (1999) Fleet assignment and routing with schedule synchronization constraints. Eur J Oper Res 119(1):75–90

Jacobs TL, Smith BC, Johnson EL (2008) Incorporating network flow effects into the airline fleet assignment process. Transp Sci 42(4):514–529

Klabjan D (2005) Large-scale models in the airline industry. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column generation. Springer, New York, pp 163–195

Rexing B, Barnhart C, Kniker T, Jarrah A, Krishnamurthy N (2000) Airline fleet assignment with time windows. Transp Sci 34(1):1–20

Sherali HD, Bish EK, Zhu X (2006) Airline fleet assignment concepts, models, and algorithms. Eur J Oper Res 172(1):1–30

Sherali HD, Bae K-H, Haouari M (2009) Integrated airline schedule design and fleet assignment: polyhedral analysis and Benders decomposition approach. INFORMS J Comput 22(4):500–513

Smith BC, Johnson EL (2006) Robust airline fleet assignment: imposing station purity using station decomposition. Transp Sci 40(4):497–516