Les Cahiers du GERAD ISSN: 0711–2440

Periodic Airline Fleet Assignment with Time Windows, Spacing Constraints, and Time Dependent Revenues

N. Bélanger, G. Desaulniers,
F. Soumis, J. Desrosiers

G–2003–41

June 2003
Revised: May 2005

Periodic Airline Fleet Assignment with Time Windows, Spacing Constraints, and Time Dependent Revenues

Nicolas Bélanger
Kronos Incorporated
nbelanger@kronos.com

Guy Desaulniers
François Soumis
GERAD and École Polytechnique
{guy.desaulniers;francois.soumis}@gerad.ca

Jacques Desrosiers
GERAD and HEC Montréal
jacques.desrosiers@hec.ca

June 2003
Revised: May 2005

Les Cahiers du GERAD

G–2003–41

Abstract

Given the sets of flights and aircraft of an airline carrier, the fleet assignment problem consists of assigning the most profitable aircraft type to each flight. In this paper we propose a model for the periodic fleet assignment problem with time windows in which departure times are also determined. Anticipated profits depend on the schedule and the selection of aircraft types. In addition, short spacings between consecutive flights which serve the same origin-destination pair of airports are penalized. We propose a non-linear integer multi-commodity network flow formulation. We develop new branch-and-bound strategies which are embedded in our branch-and-price solution strategy. Finally, we present computational results for periodic daily schedules on three real-world data sets.

Key Words: Integer programming; Dantzig-Wolfe decomposition; Column generation; Routing; Scheduling; Airline fleet assignment.

Résumé

Le problème d'affectation des types d'avions aux vols consiste à déterminer le choix le plus profitable pour un transporteur aérien quant au type d'avion à être affecté à chacun des vols d'un horizon donné. Nous proposons dans cet article une extension avec fenêtres de temps sur les heures de départ dans laquelle est pénalisé l'espacement trop court entre les vols desservant consécutivement la même paire de villes. Les profits sont estimés en fonction des heures de départ choisies et des types d'avions sélectionnés. Nous proposons un modèle multi-flots non linéaire en nombres entiers intégrant des variables de temps. L'approche de résolution fait appel à la génération de colonnes et de nouvelles stratégies de branchement ont été développées. Nous présentons les résultats d'un ensemble de tests numériques pour un horizon périodique d'une journée réalisés sur trois jeux de données.

Mots clés : Programmation en nombres entiers; décomposition Dantzig-Wolfe; génération de colonnes; routage; horaire; transport aérien.

1 Introduction

Given the set of flights and the aircraft fleet of an airline carrier, the fleet assignment problem consists of assigning the most profitable aircraft type to each flight leg. An extension to this problem is to allow for some flexibility on flight departure times. These time adjustments increase the number of possible connections and therefore improve the solution quality in terms of profits or in the reduction of the number of aircraft utilized to cover all flights. However, such adjustments should be done with care. For instance, if the departure times of two flights which serve the same pair of cities become too close, these flights compete with each other and passenger demands on these flights may be overestimated. Since passenger demand depends on departure times, estimates of flight revenues may as well be inaccurate when relatively large time windows are used. Therefore it is important to control the spacing out between consecutive flights which serve the same origin-destination pair and the demand estimations within the time windows.

In this paper we consider a periodic scheduling horizon and we propose a model for the fleet assignment problem with time windows in which short spacings between consecutive flights which serve the same origin-destination pair are penalized while profit estimates are functions of both departure times and aircraft types. We propose a mathematical model based on a non-linear integer multi-commodity network flow structure. We also present a solution approach using a branch-and-price strategy applied to a decomposition of the model. Columns are generated using a specialized longest path problem with time windows and linear costs on arc flows and on node visiting times. New branching strategies are presented, which appear to be essential for efficiently solving the problem. Our experiments show that this approach performs well for solving daily instances with large time windows.

The paper is organized as follows: In Section 2 we describe the problem. The literature is reviewed in the following section. A mathematical model is developed in Section 4. The column generation solution approach is presented in Section 5 followed in Section 6 by a detailed description of the branch-and-bound strategies. Section 7 presents a series of results for periodic daily schedules on three real-world data sets. Our conclusions are reported in Section 8.

2 Problem Description

A flight is characterized by a pair of origin-destination (O-D) airports, its duration with respect to an aircraft type and a time window in which departure must occur. Given a heterogeneous fleet of aircraft and a scheduling horizon, we are looking for the best assignment of aircraft types to flights together with the departure time of each flight. Restrictions on time windows may apply for some aircraft, for example, to respect curfew imposed at certain airports. An aircraft itinerary is a sequence of flights (that can be covered by a single aircraft) between consecutive O-D airports for which departure times within the corresponding time windows allow for sufficient connection times. We assume here that the minimum connection time depends on the inbound flight and the aircraft type. In some cases, it may also depend on the outbound flight. Indeed, a domestic flight followed by an international one (or vice versa) may require the towing of the aircraft between assigned gates, and hence may significantly increase the connection time.

In a feasible fleet assignment the itineraries of all aircraft must be constructed in such a way that each flight is operated by the selected type, and that the itineraries can be repeated over the subsequent scheduling horizons. An optimal assignment maximizes the total anticipated profits, taking into account the negative impact of short spacing between the departure times of consecutive flights as well as the fixed costs incurred by the use of the aircraft fleet. In this paper, the anticipated profits of a flight do not only depend on the aircraft type but also on the selected departure time which in turn influences passenger demand. Moreover, in order to minimize the fleet size, we use a large fixed cost for the utilization of an aircraft.

3 Literature Review

Literature on the fleet assignment problem with a fixed schedule spans over several decades, the most recent papers being those of Abara (1989), Subramanian *et al.* (1994), Hane *et al.* (1995), and Rushmeier and Kontogiorgis (1997). Formulations and solution approaches are very similar. They rely on a mixed integer multi-commodity network flow formulations based on a space-time graph representations that are solved by branch-and-bound.

Richer fleet assignment models taking into account additional components of the airline decision process have also been considered. Clarke *et al.* (1996) introduce maintenance and crew considerations while Barnhart *et al.* (1998) propose an extended weekly model with maintenance constraints that is solved using a branch-and-price approach. Each node of the search tree corresponds to the linear relaxation of a problem based on a set partitioning model solved by column generation where the column generator is a longest path problem. Columns in the set partitioning problem refer to feasible aircraft itineraries.

Two types of approaches have been proposed in the literature to handle departure time windows. In the first (Levin (1971) and Rexing *et al.* (2000)), time windows are discretized, hence creating a number of copies for each flight in the underlying space-time network representation. The problem then formulates as an integer program that is solved by a branch-and-bound algorithm or an iterative solution approach proposed in Rexing *et al.* (2000). In the second (Desaulniers *et al.* (1997)), a multi-commodity network flow model involving continuous time variables and time window restrictions on these variables is solved by branch-and-price where the column generator is a time constrained longest path problem (see Desrochers and Soumis, 1988). The first approach type has the disadvantage of being less accurate than the second type when a coarse discretization is used to obtain a tractable formulation size. However, it can easily handle time dependent revenues which require an increasing complexity of the column generator in the second approach type. For both types, similar constraints must be incorporated into the model to account for spacing requirements. In this paper, we chose an approach of the second type due to its higher accuracy.

Bélanger *et al.* (2002) address the fleet assignment problem for a weekly flight schedule where it is desirable to assign the same type of aircraft to the legs operating with the same flight number on different days of the week. Even though it may reduce schedule profitability, aircraft type homogeneity is sought in order to improve the planning and implementation of the operations.

A solution to the fleet assignment problem must maximize the total expected profits from the flight schedule. In our model, expected profits are computed by flight leg and vary according to the aircraft type assigned to it. This is an approximation of the reality since passenger demand arises by O-D pair and the passenger demand for each flight leg can only be obtained when passenger routing is known. However, this routing depends on the capacity of each leg given by the fleet assignment. In order to get a better approximation, our model could be integrated in a feedback loop with a passenger routing model. Another alternative, introduced in Barnhart *et al.* (2002), would be to consider simultaneous fleet assignment and passenger routing. However, the work of these authors does not address the bi-level optimization problem involved in the interaction between the airline that tries to maximize their profits, and the passengers that select their routing among the choices proposed by the airline and its competitors according to their preferences (namely, a low ticket fare). Further research is, therefore, needed to obtain a more realistic model that includes this important aspect of the problem.

The model proposed in this paper for the periodic fleet assignment problem with time windows follows the general vehicle routing and crew scheduling framework proposed in Desaulniers *et al.* (1998). Multi-commodity flows, branch-and-bound and column generation are used to solve it. One difficulty is the generation of the columns, or equivalently, the feasible aircraft itineraries. To account for flight spacing constraints as well as time dependent profit estimates, itineraries are generated using a specialized time window constrained longest path problem in which the objective function is of the form $\mathbf{c}^\top\mathbf{x} + \alpha^\top\mathbf{t}$, that is, a linear combination of the arc flow ($\mathbf{x}$) and node time ($\mathbf{t}$) variables. Ioachim *et al.* (1998) propose an efficient dynamic programming algorithm to solve that type of time constrained path problem. It has already been used in several applications, among which are aircraft routing with schedule synchronization (Ioachim *et al.*, 1999) and simultaneous optimization of flight and pilot schedules in a recovery environment (Stojković and Soumis, 2001). In the former application, flights on certain O-D pairs must be scheduled at the same time but on different days of a weekly horizon. In the latter, small schedule perturbations do not alter aircraft itineraries but flight departure times are delayed at a certain cost.

4 A Mathematical Model

Let K and L be the sets of aircraft and flight legs, respectively. Assume the scheduling horizon is given by the interval $[0, T]$. For a flight l operated by aircraft k, let d_l^k and $[A_l^k, B_l^k]$ be its duration and the departure time window, respectively. The duration of a flight is defined as the time elapsed between its departure time and the moment the aircraft is ready for a subsequent flight, that is, it includes a minimal connection time at the arrival airport. In a periodic schedule, it is allowed that time windows intersect with the horizon boundary. Without loss of generality, we assume $0 \leq A_l^k < T$ while $B_l^k > T$ is legitimate. We denote by p_l^k the anticipated profits if the departure time occurs at time A_l^k. Let λ_l^k be the slope of the profit adjustment within time window $[A_l^k, B_l^k]$. Therefore, the anticipated profits are given by the affine function $p_l^k + \lambda_l^k(t_l^k - A_l^k)$, where $t_l^k \in [A_l^k, B_l^k]$ is the optimized departure time. Additionally, for a pair (i, j) of consecutive flights $i, j \in L$ which serve the same O-D pair, a penalty P_{ij} is incurred if the spacing between consecutive departure times is less than a certain threshold denoted by Δ_{ij}. Let M be the set of such flight pairs for which this is required.

The mathematical model is built on a space-time graph which describes the possible movements of the fleet of aircraft. The following two subsections provide a description of the underlying network structure and the proposed mathematical formulation.

4.1 A Multi-Commodity Network Structure

To each aircraft $k \in K$, we assign a network flow commodity. We denote by $L^k \subset L$ the set of flights that can be served by this specific aircraft k. A space-time network $G^k = (N^k, A^k)$ is constructed, where N^k and A^k are the sets of nodes and arcs, respectively. To model the flexibility on flight departure times, time intervals on nodes and durations on arcs are required. Unless otherwise specified, all arcs of G^k are assigned a null profit and a null duration. We first define a source node o^k and a sink node d^k with time windows $(-\infty, 0]$ and $[0, T]$, respectively (see Figure 1 where $T = 1440$). Arc $o^k \to d^k$ allows for the non utilization of aircraft k.

For each flight $l \in L^k$ which serves a given O-D pair of airports the sequence $O_l^k \to n_l^k \to D_l^k$ of nodes and arcs is created. We assign time windows $[0, T]$ to the origin node O_l^k, $[A_l^k, B_l^k]$ to the flight node n_l^k, and $[0, T]$ to the destination node D_l^k. The first arc is assigned the anticipated profits p_l^k while the second arc is assigned the flight duration d_l^k.

To construct a periodic solution, we need additional nodes and arcs to ensure that all aircraft will be at the right place and time at the beginning of the next period. Depending on the departure time, a part of flight l may fall into the next scheduling period if $B_l^k + d_l^k > T$. Three new nodes are created for such a flight: flight nodes $\overline{n}_l^k$ and $\hat{n}_l^k$, to which we assign time windows $[A_l^k, B_l^k]$ and $[A_l^k - T, B_l^k - T]$, respectively; and a destination node $\hat{D}_l^k$ with time window $[0, T]$. To ensure periodicity in the mathematical formulation, the scheduling times at nodes $\overline{n}_l^k$ and $\hat{n}_l^k$ must be synchronized, that is, the time at node $\overline{n}_l^k$

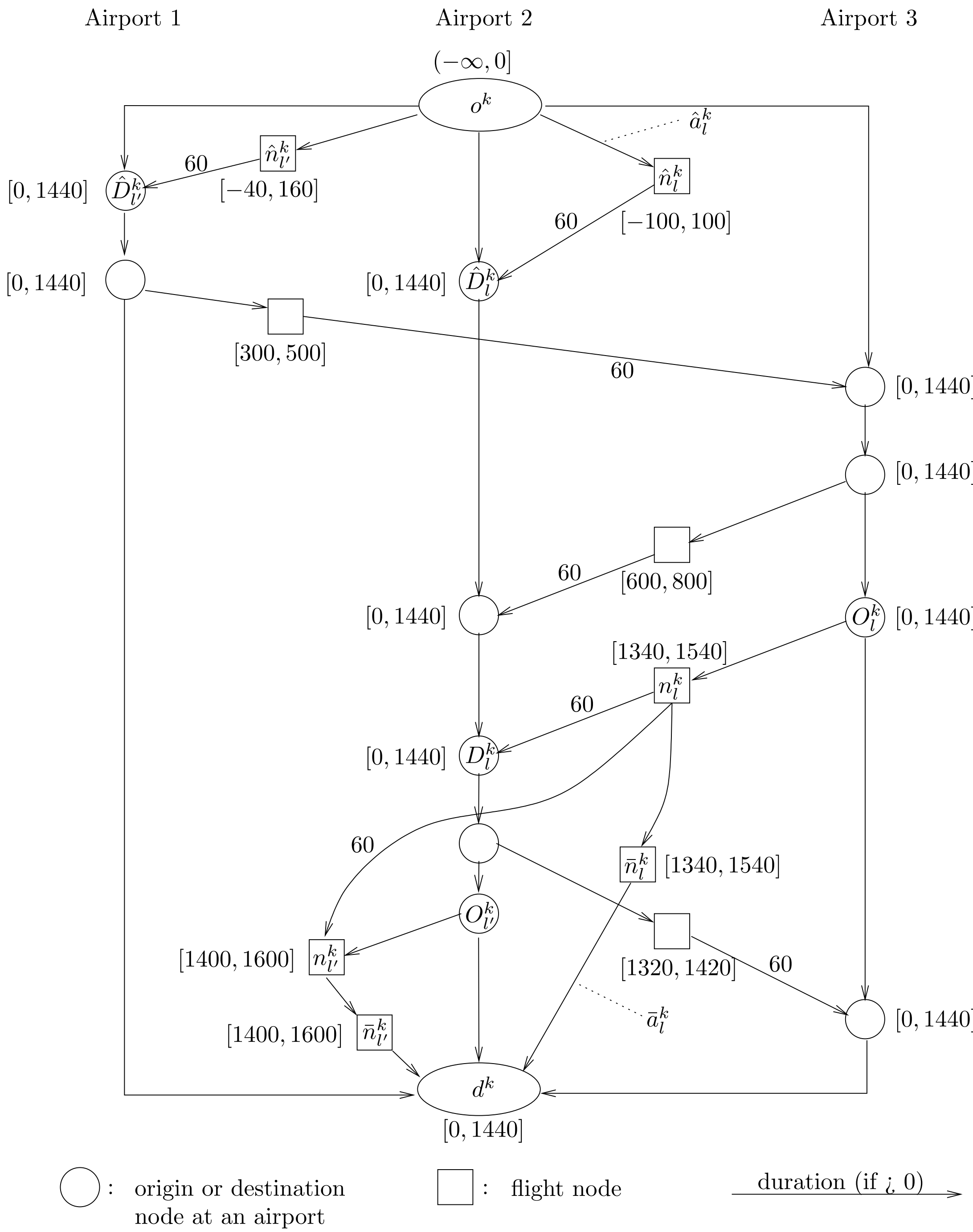

Figure 1: Network structure G^k for aircraft $k \in K$

must be equal to the time at node $\hat{n}_l^k$ plus T. The new arc sequences are: $n_l^k \rightarrow \overline{n}_l^k \rightarrow d^k$ at the end of the horizon, and $o^k \rightarrow \hat{n}_l^k \rightarrow \hat{D}_l^k$ at the beginning of the horizon. In the following, arc $\overline{n}_l^k \rightarrow d^k$ is denoted $\overline{a}_l^k$ and arc $o^k \rightarrow \hat{n}_l^k$ is denoted $\hat{a}_l^k$. Arc $\hat{n}_l^k \rightarrow \hat{D}_l^k$ is assigned the flight duration value d_l^k. Note that if $A_l^k + d_l^k > T$, the original destination node D_l^k and the arc $n_l^k \rightarrow D_l^k$ can be removed since no other flights can be served by that aircraft after flight l.

With each origin or destination node O_l^k, D_l^k, or $\hat{D}_l^k$, for all $k \in K$ and for all $l \in L^k$, is associated the corresponding takeoff or landing airport $s_l^k \in S$. For a given airport, time values are assigned to these nodes so that they can be chronologically sorted. Origin node O_l^k is assigned the latest possible departure time B_l^k while destination nodes D_l^k and $\hat{D}_l^k$ are assigned the earliest arrival time $A_l^k + d_l^k$. In case of a tie, destination nodes precede origin nodes. Ground arcs are created between the successive nodes at an airport. For $s \in S$, arc a_{os}^k links the source node o^k to the first node of airport s with a negative profit corresponding to the fixed cost incurred for using aircraft k while arc a_{ds}^k connects the last node of airport s to the sink d^k.

Remark 1. Two types of possible connections are not considered in the above basic network construction. Type 1 occurs when a flight l can end after time T and another flight l' can be served immediately after it, see Figure 2. Formally, the necessary and sufficient conditions for such a connection are: $A_l^k + d_l^k \leq B_{l'}^k$, $B_l^k + d_l^k > T$, and $B_{l'}^k > T$. In this case, arc $n_l^k \rightarrow n_{l'}^k$ is created (see Figure 1) to which we assign the duration d_l^k and the anticipated profits $p_{l'}^k$.

Connections of type 2 appear when a flight l' can start after time T (i.e., $B_{l'}^k > T$) following a flight l starting just after T, that is, at the beginning of the next horizon (see Figure 3). The conditions are: $A_l^k + d_l^k \leq B_{l'}^k - T$. Flight l is located at the beginning of the horizon while flight l' is at the end, which forbids the allowed connection. In this case, two supplementary nodes are created for l': $\tilde{n}_{l'}^k$ with time window $[A_{l'}^k - T, B_{l'}^k - T]$ and $\tilde{D}_{l'}^k$ with time window $[0, T]$. The arc sequence is $n_l^k \rightarrow \tilde{n}_{l'}^k \rightarrow \tilde{D}_{l'}^k$; the first arc supports the flight duration d_l^k and the anticipated profits $p_{l'}^k$ while the second arc is assigned the flight duration $d_{l'}^k$.

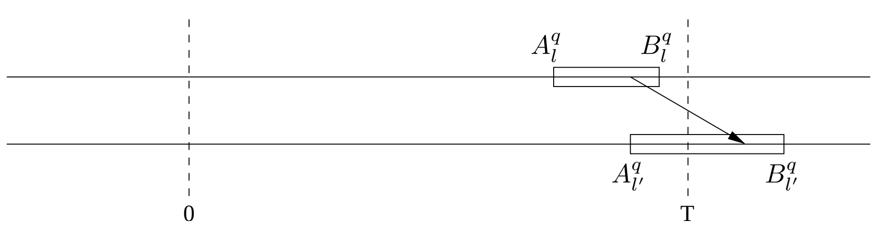

Figure 2: Type 1 connection

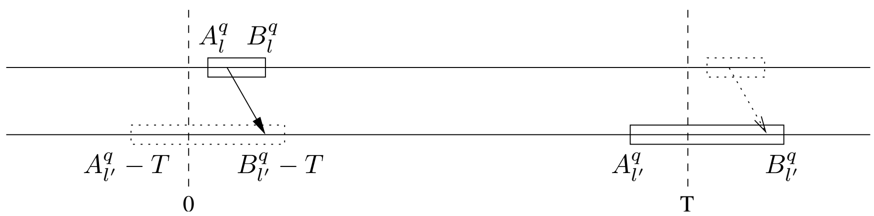

Figure 3: Type 2 connection

4.2 A Mathematical Formulation

Given a node $n \in N^k$, $k \in K$, let $\Gamma(n) \subset A^k$ be the subset of outgoing arcs and $\Gamma^{-1}(n) \subset A^k$ the set of entering arcs. The associated time window is denoted $[a_n^k, b_n^k]$. A linear cost $\alpha_n^k = \lambda_l^k$ is assigned to each node $n = n_l^k$ and $n = \hat{n}_l^k$, $l \in L^k$, and 0 to all the remaining nodes. We assign a profit c_a^k and a duration d_a^k to each arc. And a binary flow variable x_a^k assumes the value 1 if and only if arc $a \in A^k$ is used. We also define the time variable t_n^k which represents the difference between the visiting time and the beginning a_n^k of the time window at node $n \in N^k$. This time definition has been chosen in place of the service time at node n for the reason of an improved numerical stability. Given two consecutive flight legs i and j, $(i,j) \in M$, we also define a non-negative time variable T_{ij} that is used to penalize a spacing between the corresponding departure times which is below a certain threshold Δ_{ij}.

To complete the notation, let $K^l \subset K$ be the subset of aircraft that can cover flight leg $l \in L$. Define Q as the set of aircraft types and let $K_q \subset K$ be the subset of aircraft of type $q \in Q$. Finally, define $P \subset L$ as the subset of flights that overlap on two periods and need to satisfy synchronization constraints in a periodic schedule.

To simplify the presentation of the mathematical model and without loss of generality, we assume that the networks are constructed over two periods so that it is not necessary to introduce the *modulo function* within several constraints which involve the difference of time variables. A mathematical formulation of the problem based on the previous definitions and the multi-commodity network structure is as follows:

$$\text{Maximize } Z = \sum_{k \in K} \sum_{a \in A^k} c_a^k x_a^k + \sum_{k \in K} \sum_{n \in N^k} \alpha_n^k t_n^k - \sum_{(i,j) \in M} P_{ij} T_{ij} \qquad (1)$$

subject to:

$$\sum_{k\in K^l}\sum_{a\in\Gamma(n_l^k)} x_a^k = 1\,, \qquad \forall l \in L \tag{2}$$

$$\sum_{k\in K^j}(t^k_{n^k_j} + a^k_{n^k_j}X^k_{n^k_j}) - \sum_{k\in K^i}(t^k_{n^k_i} + a^k_{n^k_i}X^k_{n^k_i}) + T_{ij} \ge \Delta_{ij}\,, \qquad \forall (i,j)\in M \tag{3}$$

$$\sum_{k\in K_q} x^k_{a^k_{os}} = \sum_{k\in K_q} x^k_{a^k_{ds}}\,, \qquad \forall q\in Q, s\in S \tag{4}$$

$$\sum_{k\in K_q\cap K^l} x^k_{\bar{a}^k_l} = \sum_{k\in K_q\cap K^l} x^k_{\hat{a}^k_l}\,, \qquad \forall q\in Q, l\in P \tag{5}$$

$$\sum_{k\in K^l}(t^k_{\bar{n}^k_l} + a^k_{\bar{n}^k_l}X^k_{\bar{n}^k_l}) = \sum_{k\in K^l}(t^k_{\hat{n}^k_l} + a^k_{\hat{n}^k_l}X^k_{\hat{n}^k_l})\,, \qquad \forall l\in P \tag{6}$$

$$\sum_{a\in\Gamma(n)} x_a^k - \sum_{a\in\Gamma^{-1}(n)} x_a^k = \begin{cases} 1, & \forall k\in K,\ n = o^k \\ 0, & \forall k\in K,\ n\in N^k\setminus\{o^k,d^k\} \\ -1, & \forall k\in K,\ n = d^k \end{cases} \tag{7}$$

$$0 \le t_n^k \le (b_n^k - a_n^k)X_n^k\,, \qquad \forall k\in K,\ n\in N^k \tag{8}$$

$$x^k_{(n,m)}(t^k_n + a^k_n + d^k_{(n,m)} - (t^k_m + a^k_m)) \le 0\,, \qquad \forall k\in K,\ n\in N^k,\ m\in\Gamma(n) \tag{9}$$

$$X_n^k = \sum_{a\in\Gamma(n)} x_a^k\,, \qquad \forall k\in K,\ n\in N^k \tag{10}$$

$$x_a^k \in \{0,1\}\,, \qquad \forall k\in K,\ a\in A^k. \tag{11}$$

The objective function (1) reflects maximization of the total anticipated profits adjusted for the spacing penalties and the fixed costs for the use of the aircraft fleet. Constraint set (2) ensures the covering of all flights exactly once. Each of the constraints in (3) is used to compute the deviation between the optimized spacing time and the threshold Δ_{ij} for two consecutive flights i and j which operate between the same O-D pair, $(i,j)\in M$. Note that we have implemented the special case where the time windows do not depend on the aircraft type, i.e., $a_n^k = a_n$, $\forall n\in N^k$, $k\in K$; therefore, the spacing constraints are rewritten as

$$\sum_{k\in K^j} t^k_{n^k_j} - \sum_{k\in K^i} t^k_{n^k_i} + T_{ij} \ge \Delta_{ij} + a_{n_i} - a_{n_j}, \quad \forall (i,j)\in M. \tag{12}$$

Constraints (4)–(6) are imposed to ensure a periodic schedule. Set (4) guarantees that the number of aircraft on the ground for each type q is the same at the beginning and at the end of the schedule horizon in each airport s. For any flight that overlaps the horizon boundaries, that is, for $l \in P$, constraints (5) ensure the flow conservation for the supplementary arcs introduced to model the presence of these overlapping flights. Set (6) synchronizes the corresponding scheduling times on the supplementary flight nodes. Note again that with $a_n^k = a_n$, $\forall n \in N^k$, $k \in K$, these synchronization constraints are rewritten as

$$\sum_{k \in K^l} t^k_{\bar{n}^k_l} = \sum_{k \in K^l} t^k_{\hat{n}^k_l}, \quad \forall l \in P. \tag{13}$$

The remaining constraints are separable by aircraft $k \in K$. Constraints (7) are classical path constraints from the source node o^k to the sink node d^k. The time window at each node is defined by (8) and it imposes a zero width if the node is not visited. In this case, there is no contribution of that time variable in the objective function, in the spacing (3) and the synchronization (6) constraints. Non-linear constraints (9) specify the time adjustment from one node to the other. Finally, (10) are used only to simplify the presentation of several expressions. The domain of the variables is given by (11). Note that it is not necessary to require integrality on time variables.

Remark 2. The binary restrictions on flow variables given by constraints (11) are not necessary to obtain a feasible solution to the problem. To determine the aircraft type assigned to a flight, it would be sufficient to impose binary restrictions on the following smaller subsets given by the sum of flow variables:

$$X^k_{n^k_l}, X^k_{\bar{n}^k_l}, X^k_{\hat{n}^k_l} \in \{0,1\}, \;\; \forall\, k \in K,\, l \in L^k. \tag{14}$$

Remark 3. If the instance to be solved involves special connections of the second type, one needs to define the variables $X^k_{\tilde{n}^k_l}$ and adequately adjust for the flight covering (2) and spacing out (3) constraints. As for the other sets of variables mentioned in Remark 2, binary restrictions must be imposed also on those variables if binary restrictions on flow variables are discarded.

5 Column Generation

Problem (1)–(11) is solved using a branch-and-price strategy. Our methodology follows the general scheme inspired by the Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960), as described in Desaulniers *et al.* (1998). Constraints (7)–(11) form a block angular structure separable for the $|K|$ groups of variables $(\mathbf{X}^k, \mathbf{T}^k)$, $k \in K$, where $\mathbf{X}^k = (x^k_a,\, a \in A^k)$ and $\mathbf{T}^k = (t^k_n,\, n \in N^k)$, corresponding to the itinerary and the schedule of an aircraft. For $k \in K$, let Ω^k be the set of extreme points of constraint set (7)–(11). Each one represents a path in network G^k. Recall that the non-linear constraints (9) can be relaxed by the following linearization (see Desrosiers *et al.*, 1995, p. 86)

$$t^k_n + d^k_{(n,m)} - t^k_m \leq (1 - x^k_{(n,m)})(max\{b^k_n + d^k_{(n,m)} - a^k_m, 0\}),\ \forall k \in K,\ n \in N^k, m \in \Gamma(n).$$

Therefore (7)–(11) can be considered as a set of linear constraints. Since the associated polyhedron is bounded, feasible points can be expressed as a convex combination of its extreme points only. The resulting expression substitutes variables in (1)–(6). For each path $p \in \Omega^k$, define a variable $\theta_p^k \in [0,1]$ and let

$$x_a^k = \sum_{p\in\Omega^k} x_{ap}^k \theta_p^k \quad \text{and} \quad t_n^k = \sum_{p\in\Omega^k} t_{np}^k \theta_p^k, \tag{15}$$

where x_{ap}^k and t_{np}^k correspond in path p to arc flow and node time variables x_a^k and t_n^k, respectively. Therefore, we can reformulate the linear relaxation of problem (1)–(11) with (3) and (6) replaced by (12) and (13), respectively, as follows:

$$\text{Maximize Z=} \sum_{k\in K}\sum_{p\in\Omega^k} c_p^k\theta_p^k - \sum_{(i,j)\in M} P_{ij}T_{ij} \tag{16}$$

subject to:

$$\sum_{k\in K}\sum_{p\in\Omega^k} v_{n_l^k p}^k \theta_p^k = 1\,, \qquad \forall l \in L \tag{17}$$

$$\sum_{k\in K}\sum_{p\in\Omega^k} \tau_{ijp}^k\theta_p^k + T_{ij} \geq \Delta_{ij} + a_{n_i^k}^k - a_{n_j^k}^k\,, \qquad \forall (i,j) \in M \tag{18}$$

$$\sum_{k\in K_q}\sum_{p\in\Omega^k} b_{sp}^k\theta_p^k = 0\,, \qquad \forall q \in Q,\ s \in S \tag{19}$$

$$\sum_{k\in K_q\cap K^l}\sum_{p\in\Omega^k} d_{lp}^k\theta_p^k = 0\,, \qquad \forall q \in Q,\ l \in P \tag{20}$$

$$\sum_{k\in K^l}\sum_{p\in\Omega^k} e_{lp}^k\theta_p^k = 0\,, \qquad \forall l \in P \tag{21}$$

$$\sum_{p\in\Omega^k} \theta_p^k = 1\,, \qquad \forall k \in K \tag{22}$$

$$\theta_p^k \geq 0\,, \qquad \forall p \in \Omega^k,\ \ k \in K \tag{23}$$

where,

$$\begin{aligned} c_p^k &= \textstyle\sum_{a\in A^k} c_a^k x_{ap}^k + \sum_{n\in N^k} \alpha_n^k t_{np}^k & \qquad b_{sp}^k &= x_{a_{os}^k p}^k - x_{a_{ds}^k p}^k \\ v_{np}^k &= \textstyle\sum_{a\in\Gamma(n)} x_{ap}^k & d_{lp}^k &= x_{\bar{a}_l^k p}^k - x_{\hat{a}_l^k p}^k \\ \tau_{ijp}^k &= I(k\in K^j)t_{n_j^k p}^k - I(k\in K^i)t_{n_i^k p}^k & e_{lp}^k &= t_{\bar{n}_l^k p}^k - t_{\hat{n}_l^k p}^k\,. \end{aligned}$$

The indicator function I assumes a value of 1 if its argument is true, and 0 otherwise.

Constraint set (22) can be aggregated by aircraft type so that the number of rows of this linear program is $|L|+|M|+|Q|\times|S|+|Q|\times|P|+|P|+|Q|$. The solution in terms of the original variables can be retrieved from (15). Assigning dual variables π_l, ϵ_{ij}, β_{sq}, ϕ_{lq}, δ_l and ψ^k to constraint sets (17) through (22), respectively, the reduced cost of variable θ_p^k is given by

$$c_p^k - \sum_{l\in L} v_{n_l^k p}^k \pi_l - \sum_{(i,j)\in M} \tau_{ijp}^k \epsilon_{ij} - \sum_{s\in S} \sum_{\{q\in Q: k\in K_q\}} b_{sp}^k \beta_{sq} - \sum_{l\in P} \sum_{\{q\in Q: k\in K_q\cap K^l\}} d_{lp}^k \phi_{lq} - \sum_{\{l\in P: k\in K^l\}} e_{lp}^k \delta_l - \psi^k.$$

This formula can be rewritten, using the previous coefficient definitions, as an affine function in terms of the arc flow and node time variables using modified coefficients $\bar{c}_a^k$ and $\bar{\alpha}_n^k$:

$$\sum_{a\in A^k} \bar{c}_a^k x_{ap}^k + \sum_{n\in N^k} \bar{\alpha}_n^k t_{np}^k - \psi^k.$$

Therefore, generating columns with positive reduced cost for a possible increase of the objective function boils down, for each $k\in K$, to satisfy constraints (7)–(11) while maximizing

$$\sum_{a\in A^k} \bar{c}_a^k x_a^k + \sum_{n\in N^k} \bar{\alpha}_n^k t_n^k - \psi^k.$$

This amounts to solving, for each $k\in K$, a time window constrained longest path problem over network G^k where the objective function depends on both the flow and the time variables. To solve it, one can use the pseudo-polynomial dynamic programming algorithm described in Ioachim *et al.* (1998), as long as time window widths guarantee that network G^k is acyclic. Otherwise, one can use an appropriate discretization of the time windows.

6 Branch-and-Bound Strategies

We have assigned a large fixed cost for the use of an aircraft so as to prioritize minimization of the fleet size. This is implemented in the following way. Given the number of aircraft $v=\sum_{k\in K}\sum_{a\in\Gamma(o^k)} x_a^k$ at the root node, we fix the number of aircraft to exactly $\lceil v\rceil$. In all our test problems, this was sufficient to find the optimal number of aircraft. If the solution was still infeasible, we would explore the next integer, and so on. Similar to vehicle routing applications involving multiple types of vehicles (or multiple depots), branching decisions can be taken on whether there is a connection between two flights or not, or on assigning a specific aircraft type to a given flight or not. In both cases, two branches are created in the search tree.

Given an integer number of aircraft and the solution at a branching node, the next node is chosen according to a modified depth-first search. Let Z_{inf} denote the best known solution (which gives a lower bound on the optimal adjusted profits) and let Z^{sup} denote the largest upper bound over all active nodes. Interval $[Z_{inf}, Z^{sup}]$ on total profits is explored according to a bisection search. The first subinterval ranges from r_1 to Z^{sup}, where $r_1 = \frac{Z^{sup}+Z_{inf}}{2}$, the second from r_2 to r_1, where $r_2 = \frac{r_1+Z_{inf}}{2}$, and so on (see Figure 4). The exploration of a branch is stopped if its upper bound is smaller than r_1. When all active nodes of the first search interval have their upper bound smaller than r_1, the exploration continues with the nodes of the second interval, that is, nodes for which the upper bound is between r_2 and r_1. This strategy is applied for the remaining intervals. Each time a better integer solution is found, the interval partitioning is recomputed. This strategy favors the search for significantly better integer solutions. The search is stopped whenever $Z^{sup} - Z_{inf} \leq \epsilon$ for a given small $\epsilon > 0$, or when the CPU time reaches a maximum value.

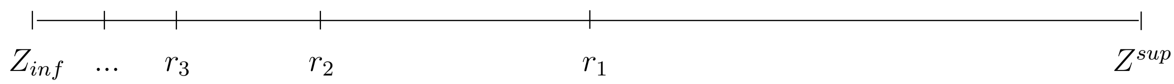

Figure 4: Bisection search in the branch-and-bound tree

Applying only the above branching strategies on our test problems was disastrous. For many of them, we could not even find a feasible solution within 50 hours of CPU time (see Section 7.2), although for several instances this was possible within only one or two hours. We carefully examined the search trees, and concluded that in many cases, we were not able to efficiently detect the infeasibility of a subtree. That is, even though we have a fractional solution available in terms of the flow variables, there are no integer solutions. Most of the time we were not even able to complete the exploration of such a subtree. We have identified two reasons for that behavior, both related to the periodicity requirements. We have implemented new branching strategies which are called before traditional ones are used.

Constraint set (13) calls for the synchronization of the time variables for a flight $l \in P$ that overlaps over the horizon boundaries. For $k \in K$, node $\bar{n}_l^k$ is located at the end of the horizon while node $\hat{n}_l^k$ is at the beginning, both being copies of flight node n_l^k. For a selected aircraft k, Bélanger (2003) constructs an example in which a fractional solution exists although no integer solutions to the problem are possible.

As a remedy to this situation, we could possibly add constraint $t^k_{\bar{n}_l^k} \geq t^k_{\hat{n}_l^k}$ in subproblem k but this would destroy the structure of the time constrained longest path problem. Additionally, this would only eliminate possible cycles within a single period and the master problem would have to manage the longer ones. Instead of that, we have implemented branching decisions on the shared time window of nodes n_l^k, $\bar{n}_l^k$ and $\hat{n}_l^k$ as follows. For a given flight $l \in P$, we first merge all identical itineraries but with flights at different departure times. The time values are computed according to the fractional flow on these itineraries. For example, if the itinerary composed of the two flights F1 and F2 {F1 (10:00), F2 (18:00)} of flow 0.2 in the solution of problem (1)–(11) is merged with the itinerary {F1 (10:21), F2 (17:51)} of flow 0.4, this gives the itinerary {F1 (10:07), F2 (17:57)} of flow 0.6. If this merging results in a single itinerary of flow 1.0,

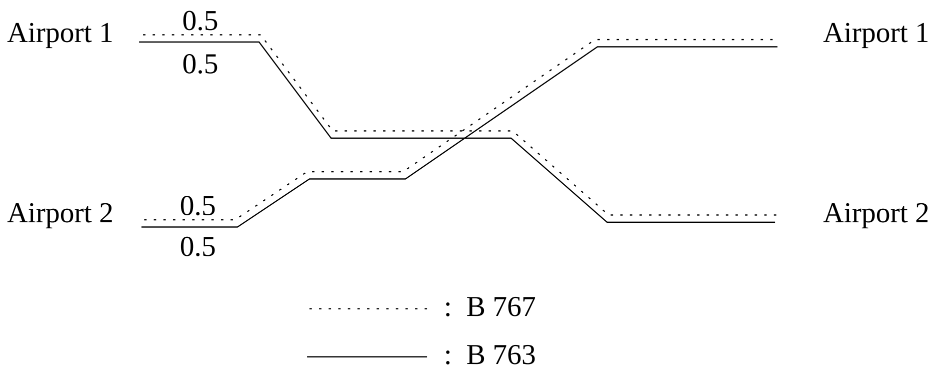

Figure 5: A fractional solution with two types of aircraft

that is, all the itineraries covering flight l are composed of the same sequence of flights, this flight is not eligible to branching on time windows. Otherwise, itineraries are sorted chronologically based upon their service time at flight l. Next we determine the largest portion that can be removed from the time window such that the first itinerary is eliminated but not the second. For example, if the time window is [10, 40] and the first itinerary serves that flight at time 11 while the second itinerary serves it at time 33, then interval [10, 32] can be removed. Assigned to this first possible decision is a score that depends on the flow on the first itinerary and the portion of the interval that is discarded. The same procedure is applied to the last itinerary, such that the branching decision removes it but not the preceding itinerary. Again a score is assigned to that second possible decision. The highest score on all possible decisions of flights in P determines the branching decision to make.

The second identified difficulty comes from the heterogeneous aspect of the fleet to achieve a periodic solution. Assume that a problem is solved on a single-day horizon and, at a certain branching node, a set of flights that makes up *a cycle over two days* has to be covered by exactly two different aircraft, based upon the previous branching decisions. Figure 5 shows two itineraries for which it is impossible to get an integer solution using aircraft types B767 and B763. However, a fractional solution with a flow of 0.5 for each aircraft on each itinerary creates two identical cycles of two days that cover all the flights exactly once. Our way to overcome this situation is twofold: On the one hand, we branch on the number of aircraft of a certain type at the end of a day, see constraint set (4), that is, on quantities $\sum_{k \in K_q} x^k_{a^k_{ds}}$ $(q \in Q,\ s \in S)$. On the other hand, we make binary decisions on the type of aircraft that covers the flights involved in the synchronization constraints, that is, on $\sum_{k \in K_q} x^k_{\bar{a}^k_l}$ $(q \in Q,\ l \in L_q)$, where $L_q \subset L$ is the subset of flights that can be covered by an aircraft of type q, a subset of variables involved in constraint set (5).

7 Computational Experiments

The solution approach has been tested using data provided by an international North-American carrier. The original data set includes a solution for a fixed flight schedule. It also gives the duration and the anticipated profits for each flight which both depend on the aircraft type. Various scenarios have been constructed to study the impact of time window flexibility and spacing.

7.1 Parameters and Statistics

Time window width is driven by a parameter μ, a percentage of the original spacing between flights offered more than once on the same O-D airports, independently of the aircraft type. Let f_i and f_j represent the fixed departure times for consecutive flights i and j, and $f_{ij} = (f_j - f_i) \bmod T$ represents the time spacing between those two departure times, $(i, j) \in M$. Then time window bounds are computed as $B_i = f_i + \mu f_{ij}$ and $A_j = f_j - \mu f_{ij}$. If a flight l is offered only once in the scheduling period, the time interval is $f_l \pm \mu T$. In both cases, the positive and negative variations are bounded from above by half the duration d_l of the flight as flown by the aircraft type given in the solution of the data set. Time windows are also limited by curfew rules at each airport.

Let t_i and t_j be the optimized departure times for consecutive flights i and j, and let $t_{ij} = (t_j - t_i) \bmod T$ be the optimized spacing, $(i, j) \in M$. By the above construction of the time windows, $t_{ij} \geq (1-2\mu)f_{ij}$, where the right-hand side is the smallest possible spacing. If the optimized spacing is shorter than a certain threshold denoted Δ_{ij}, a penalty P_{ij} is incurred. Δ_{ij} is chosen according to parameter $\nu \in [0,\ 1]$ and is computed as a convex combination of the minimum and original spacings:

$$\Delta_{ij} = (1 - 2\mu)f_{ij}\nu + f_{ij}(1 - \nu) = (1 - 2\mu\nu)f_{ij}.$$

When $\nu = 100\%$, the allowable spacing reduction incurs no cost while 75% of it is penalized if $\nu = 25\%$. The penalty term $P_{ij}T_{ij}$ becomes effective in the objective function for values $\Delta_{ij} - t_{ij} > 0$. In our tests, $P_{ij} = \frac{10^5}{f_{ij}}$, $(i, j) \in M$, that is, a one thousand penalty per minute of violation for an original spacing of 100 minutes. Recall that there is no minimum spacing when a flight is offered only once.

Revenues for a flight depend on the demand which in turn depends on its scheduled time. Profit variations λ_l^k for flight leg l on aircraft k have been determined independently of the aircraft type according to a Gaussian distribution $N(0, {\sigma_l}^2)$, where $\sigma_l = \frac{1000}{d_l}$. Finally, a large fixed cost has been utilized for the use of any aircraft in order to first minimize the fleet size, and then the adjusted profits, in that order.

Table 1 presents statistics for three flight sets. The first two data sets involve a large number of international flights while the third essentially consists of domestic and trans-border flight legs. For each data set, we give the number of airports and flight legs, the fleet size and the number of aircraft types, the number of spacing constraints, the number of overlapping flights, and finally, the row size of the linear program (16)–(23). The actual LP size varies dynamically with the introduction and the removal of constraints derived from the branch-and-bound strategies. As most of the flights are within North America, the horizon boundary has been set at $-9{:}00$ UTC so as to minimize the number of synchronization constraints in (5) and (6). Table 2 presents the average and the standard deviation of the time window widths (in minutes) as a function of the μ-parameter. On average for $\mu = 8\%$, the time intervals are as large as one hour and a half.

Table 1: Statistics on test problems

Data set	Airports	Flights	Aircraft	Aircraft types	Spacing constraints	Overlapping flights	LP row size
I	45	283	93	13	192	30	1493
II	62	411	114	13	307	32	1985
III	47	412	90	7	345	0	1093

Table 2: Average (standard deviation) of the time windows widths (in minutes)

	Time window parameter μ (%)			
Data set	2	4	6	8
I	23.7 (21.5)	47.8 (43.1)	70.8 (63.7)	89.6 (81.0)
II	22.4 (20.1)	45.2 (40.3)	66.0 (59.1)	82.8 (74.9)
III	19.5 (17.1)	38.3 (32.5)	54.0 (45.1)	63.9 (51.7)

7.2 Results

Tests have been conducted on a Sun Enterprise 10000 (400 MHz) using a single processor with a time limit of 8 hours, for parameter $\mu = 0\%, 2\%, 4\%, 6\%$, and 8%, and for parameter $\nu = 100\%, 50\%, 33\%$, and 25%. Table 3 shows the average compression factor in percentage of the original spacing between consecutive flights serving the same O-D airports. The smallest average compression is 91.4%. One can observe that as ν decreases, the optimized spacing remains closer to its original value.

The * symbol indicates the single unsolved instance. Without the three new branch-and-bound strategies developed specifically for this application, we found solutions for all 17 test problems of data set III. However, we were not able to find any integer solutions within 50 hours of CPU time for 12 instances of data set I and 10 instances of data set II. These two sets involve a large number of overlapping flights and therefore a large number of synchronization constraints (see Table 1).

As shown in Table 4, time window flexibility on departure times has a spectacular impact on the reduction of the number of aircraft needed to cover all the flights. This is

Table 3: Average compression factor (%) of the original spacings

		Time window parameter μ (%)				
Data set	ν (%)	0	2	4	6	8
I	100	100	97.6	95.7	94.0	*
	50		98.4	96.9	95.2	94.1
	33		99.0	98.9	96.6	95.4
	25		99.2	98.3	97.5	96.3
II	100	100	97.7	95.8	93.7	92.4
	50		98.4	96.9	95.3	94.4
	33		99.0	97.8	96.7	95.7
	25		99.1	98.7	97.4	96.5
III	100	100	97.8	95.7	93.2	91.4
	50		98.5	96.9	95.4	94.3
	33		99.0	97.8	96.7	95.9
	25		99.2	98.2	97.5	96.7

Table 4: Number of aircraft used

	Time window parameter μ (%)				
Data set	0	2	4	6	8
I	88	83	82	80	79
II	110	106	105	103	102
III	90	87	86	83	80

usually coupled with an increase in the anticipated (adjusted) profits, or a small decrease in some few cases. This can be seen in Table 5 where we also give the spacing penalty as a percentage of the profits computed using $\mu = 0\%$ and $\nu = 100\%$ (in parenthesis). Recall that there is no penalty when $\nu = 100\%$.

As expected, an increasing flexibility in departure times and a tightening of the spacing constraints make the problems more difficult to solve. It is relatively fast to reach a solution of the linear relaxation while it may take hours to obtain the first integer solution. For the same amount of CPU time, the average optimality gaps are also larger. Tables 6 and 7 provide results on these aspects. As seen before, the introduction of spacing and synchronization constraints in the master problem generates a number of difficulties in the branch-and-bound phase. See Bélanger (2004) for an example showing why there exists an integrality gap when using spacing constraints.

With more than 95% of the total CPU time, the solution of the master problem largely dominates the time spent to solve the column generation subproblem. Due to the presence of time constraints, the generated columns are represented not only by binary coefficients

Table 5: Percentage of profit improvements (spacing penalty)

Data set	ν (%)	Time window parameter μ (%)				
		0	2	4	6	8
I	100	1 180 693 $	25.5	29.1	28.1	*
	50		25.3 (0.0)	30.6 (0.0)	21.4 (2.4)	29.8 (0.5)
	33		24.7 (0.2)	30.0 (0.4)	22.4 (1.2)	26.3 (2.4)
	25		24.5 (0.5)	30.0 (0.7)	22.3 (1.0)	25.3 (3.7)
II	100	1 654 107 $	2.8	3.4	1.4	8.9
	50		2.4 (0.0)	3.1 (0.3)	2.4 (0.5)	8.4 (0.1)
	33		−0.5 (0.2)	2.2 (0.4)	−1.1 (1.4)	6.1 (1.7)
	25		−1.2 (0.4)	1.4 (1.0)	−3.5 (1.4)	4.9 (2.6)
III	100	1 112 004 $	8.0	11.9	13.3	13.2
	50		7.4 (0.1)	11.4 (0.1)	9.4 (0.6)	12.2 (0.2)
	33		6.9 (0.4)	9.4 (1.4)	8.7 (2.3)	7.2 (1.8)
	25		6.6 (0.6)	6.8 (2.6)	8.4 (2.4)	7.9 (2.5)

corresponding to the covered flights but also by their departure times. This makes the master problem much larger in row size compared to the usual set partitioning formulation. Moreover, the spacing constraints introduce coefficients of very different scales.

One line of future research in this area is certainly the acceleration of the master problem solution. Perturbation of the right-hand side of the synchronization constraints might be of interest (see Desaulniers *et al.*, 1999).

8 Conclusion

We have proposed an extension of the periodic aircraft fleet assignment problem that incorporates time dependent revenues within time windows on departure times as well as penalties to account for short spacings between consecutive flights serving the same O-D airports. This formulation is solved using a branch-and-price approach where the master problem involves spacing and synchronization constraints added to the classical set partitioning model. Columns represent aircraft itineraries and are generated using a specialized time constrained longest path subproblem using arc cost for flow variables and node cost for time variables.

Computational experiments conducted on data sets provided by a medium North-American airline carrier show that classical branch-and-bound strategies are not sufficient. We have introduced three new strategies which are specific to this application. Time windows allow for an important reduction in the number of aircraft, most of the time accompanied by increasing profits. The incorporation of penalized time variables in addition to the usual network flow variables permits to adequately control the spacing restrictions.

Table 6: CPU times (min.) to find the LP (first integer) solution

		Time window parameter μ (%)				
Data set	ν (%)	0	2	4	6	8
I	100	2 (2)	3 (15)	4 (193)	3 (45)	4 (*)
	50		5 (82)	7 (92)	8 (135)	8 (161)
	33		6 (75)	8 (76)	8 (122)	9 (136)
	25		6 (64)	9 (74)	9 (133)	8 (152)
II	100	5 (5)	8 (84)	8 (66)	8 (162)	9 (143)
	50		16 (188)	21 (294)	19 (373)	20 (368)
	33		20 (228)	23 (320)	23 (355)	26 (358)
	25		18 (259)	23 (295)	24 (366)	26 (327)
III	100	4 (26)	5 (5)	4 (6)	4 (114)	5 (9)
	50		17 (130)	16 (152)	22 (230)	20 (282)
	33		18 (174)	20 (112)	21 (147)	22 (174)
	25		16 (162)	21 (98)	20 (202)	22 (147)

Table 7: Optimality gaps (%)

		Time window parameter μ (%)				
Data set	ν (%)	0	2	4	6	8
I	100	0	0.3	1.9	1.0	*
	50		0.3	0.6	6.5	3.5
	33		0.6	0.9	5.3	5.8
	25		0.6	0.5	5.0	6.1
II	100	0	0.1	0.6	4.4	1.3
	50		0.4	0.7	3.0	1.5
	33		2.9	1.2	6.0	3.2
	25		3.5	1.7	8.1	3.9
III	100	0	0.0	0.0	0.6	0.0
	50		0.2	0.1	3.6	0.4
	33		0.2	1.2	3.3	4.1
	25		0.2	3.1	2.5	2.2

References

Abara, J. (1989), "Applying Integer Linear Programming to the Fleet Assignment Problem", *Interfaces* 19, 20–28.

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

Barnhart, C., Kniker, T.S., and Lohatepanont, M., "Itinerary-Based Airline Fleet Assignment", *Transportation Science* 36(2), 199–217, 2002.

Bélanger, N. (2004), *Planification d'une flotte d'avions*, Ph.D. dissertation, École Polytechnique de Montréal.

Bélanger, N., Desaulniers, G., Soumis, F., Desrosiers, J., and Lavigne, J. (2005), "Weekly Airline Fleet Assignment with Homogeneity", *Transportation Research* **B**.

Clarke L.W., Hane C.A., Johnson E.L., and Nemhauser G.L. (1996), "Maintenance and Crew Considerations in Fleet Assignment", *Transportation Science* 30(3), 249–260.

Dantzig, G.B. and Wolfe P. (1960), "Decomposition Principle for Linear Programs", *Operations Research* 8, 101–111.

Desaulniers G., Desrosiers J. , Dumas Y., Solomon M.M., and Soumis F. (1997), "Daily Aircraft Routing and Scheduling", *Management Science* 43, 841–855.

Desaulniers G., Desrosiers J., Ioachim I., Solomon M.M., Soumis F., and Villeneuve D. (1998), "A Unified Framework for Time Constrained Vehicle Routing and Crew Scheduling Problems", *Fleet Management and Logistics*, T. Crainic and G. Laporte (eds), Kluwer, Norwell, MA, 57–93.

Desaulniers G., Desrosiers J., and Solomon M.M. (2002), "Accelerating Strategies in Column Generation Methods for Vehicle Routing and Crew Scheduling Problems", in C.C. Ribeiro and P. Hansen (eds), *Essays and Surveys in Metaheuristics*, Kluwer, Norwell, MA, 309-324.

Desrosiers J., Dumas Y., Solomon M.M., and Soumis F. (1995), "Time Constrained Routing and Scheduling", Network Routing: *Handbooks in Operations Research and Management Science* 8, M. Ball, T. Magnanti, C. Monma, G. Newhauser (eds.), Elsevier Science Publisher B.V., Amsterdam, 35–139.

Hane C., Barnhart C., Johnson E.L., Marsten R.E., Nemhauser G.L., and Sigismondi G. (1995), "The Fleet Assignment Problem: Solving a Large-Scale Integer Program", *Mathematical Programming* 70, 211-232.

Ioachim I., Desrosiers J., Soumis F., and Bélanger N. (1999), "Fleet Routing with Schedule Synchronization Constraints", *European Journal of Operational Research* 119, 75–90.

Ioachim I., Gélinas S., Desrosiers J., and Soumis F. (1998), “A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs”, *Networks* 31, 193–204.

Levin, A. (1971), “Scheduling and Fleet Routing Models for Transportation Systems.” *Transportation Science* 5, 232–255.

Rexing B., Barnhart C., Kniker T., Jarrah A., and Krishnamurthy N. (2000), “Airline Fleet Assignment with Time Windows”, *Transportation Science* 34, 1–20.

Rushmeier, R.A. and Kontogiorgis S.A. (1997), “Advances in the Optimization of Airline Fleet Assignment”, *Transportation Science* 31, 159–169.

Stojković M. and Soumis F. (2001), “An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem”, *Management Science* 47, 1290–1305.

Subramanian R., Scheff R.P., Quillinan J.D., Wiper D.S., and Marsten R.E. (1994), “Coldstart: Fleet Assignment at Delta Air Lines”, *Interfaces* 24, 104–120.