

**Daily Aircraft Routing and  
Scheduling**

G. Desaulniers, J. Desrosiers  
Y. Dumas, M. M. Solomon  
F. Soumis

G-94-21

May 1994  
Revised: August 1995

Les textes publiés dans la série des rapports de recherche HEC n'engagent que la responsabilité de leurs auteurs. La publication de ces rapports de recherche bénéficie d'une subvention du Fonds F.C.A.R.

# Daily Aircraft Routing and Scheduling

**Guy Desaulniers**

GERAD and École Polytechnique de Montréal  
Montréal, Québec, Canada, H3C 3A7

**Jacques Desrosiers**

GERAD and École des Hautes Études Commerciales  
Montréal, Québec, Canada, H3T 1V6

**Yvan Dumas**

Ad Opt Technologies  
Montréal, Québec, Canada, H3V 1G4

**Marius M. Solomon**

Northeastern University and GERAD  
Boston, Massachusetts, U.S.A., 02115

**François Soumis**

GERAD and École Polytechnique de Montréal  
Montréal, Québec, Canada, H3C 3A7

June 30, 1994

Revised: August, 1995

## Abstract

In this paper we consider the daily aircraft routing and scheduling problem (DARSP). It consists of determining daily schedules which maximize the anticipated profits derived from the aircraft of a heterogeneous fleet. This fleet must cover a set of operational flight legs with known departure time windows, durations and profits according to the aircraft type. We present two models for this problem: a Set Partitioning type formulation and a time constrained multi-commodity network flow formulation. We describe the network structure of the subproblem when a column generation technique is applied to solve the linear relaxation of the first model and when a Dantzig-Wolfe decomposition approach is used to solve the linear relaxation of the second model. The linear relaxation of the first model provides lower bounds. Integer solutions to the overall problem are derived through branch-and-bound. By exploiting the equivalence between the two formulations, we propose various optimal branching strategies compatible with the column generation technique. Finally we report computational results obtained on data provided by two different airlines. These results show that significant profit improvement can be generated by solving the DARSP using our approach and that this can be obtained in a reasonable amount of CPU time.

## Résumé

Dans cet article, nous étudions le problème de fabrication quotidienne des itinéraires et des horaires des avions d'une flotte hétérogène afin de maximiser les profits anticipés d'une compagnie aérienne. Ces itinéraires doivent couvrir un ensemble de segments de vol opérationnels dont les fenêtres de temps de départ, les durées de vol et les profits anticipés sont connus en fonction de chaque type d'avion. Nous formulons ce problème de deux façons différentes, soit comme un problème de partition d'un ensemble avec contraintes supplémentaires et comme un problème de multiflot avec contraintes de temps. Nous décrivons la structure du réseau du sous-problème lorsque la relaxation linéaire est résolue par une technique de génération de colonnes pour le premier modèle ou par une approche de décomposition de Dantzig-Wolfe pour le second modèle. Les solutions entières sont obtenues en utilisant un algorithme de séparation et d'évaluation progressive. Les bornes inférieures sont calculées en résolvant la relaxation linéaire du problème. En exploitant l'équivalence entre les deux formulations, nous proposons plusieurs stratégies de branchement optimales compatibles avec la technique de génération de colonnes. Finalement, nous présentons certains résultats numériques obtenus à partir de données fournies par deux compagnies aériennes. Ces résultats montrent que ce problème peut être résolu en un temps raisonnable par une technique de génération de colonnes apportant ainsi une augmentation substantielle des profits.

# Contents

1	Introduction	1
2	A Set Partitioning Type Formulation	5
3	The Subproblem Network Structure	6
4	A Time Constrained Multi-Commodity Network Flow Formulation	11
5	Branch-and-Bound Strategies	13
6	Extensions	15
7	Computational Results	19
8	Conclusions	25
9	References	26

# 1 Introduction

In this paper, we consider the problem faced by an airline needing to construct daily schedules for a heterogeneous aircraft fleet. An aircraft schedule consists of a sequence of flight legs to be carried out by an aircraft (the routing aspect) and the exact times at which these flight legs must start and end (the scheduling aspect). This daily aircraft routing and scheduling problem (DARSP) is very important for an airline since a fleet schedule determines a large part of the airline cost estimates including the fixed cost for each aircraft, the cost of the fuel consumed and the salaries of the crew members. Furthermore, the total revenue of the airline derived from a given fleet schedule can also be estimated if a demand function is known for each flight leg to be flown.

Obviously, different aircraft schedules lead to different costs and revenues for the airline. For example, a flight leg that can be flown by two aircraft of different capacities can result in a loss of revenue if the smaller aircraft is chosen when the demand for the leg exceeds the smaller aircraft's capacity. There can be a significantly larger difference in profits between two daily schedules that require different aircraft fleet sizes since the fixed cost of an aircraft is very high. Therefore, substantial cost reductions and revenue increases can be achieved by optimally solving the DARSP. Such a superior profit position is an immediate necessity for any airline participating in today's extremely competitive market. Nevertheless, as the DARSP is a highly complex problem, its optimal solution and the benefits stemming from it cannot be achieved without the use of sophisticated mathematical programming tools as the ones we are about to develop in this paper.

**Problem Description:** The definition of the DARSP varies across the literature because many factors influence the costs and revenues of a fleet schedule and numerous types of constraints can be taken into account while constructing such a schedule. An extended version of the problem can include the selection of the flight legs to fly, their frequency and departure times, as well as the collective agreements which regulate the salaries of the crew members and, consequently, affect the costs of a fleet schedule. Unfortunately, such a version is very unlikely to be solved even for small airlines since the combinatorial aspect of the problem plays a highly significant role. In this paper, we will consider the following definition of DARSP:

*Given a heterogeneous aircraft fleet, a set of operational flight legs over a one-day horizon, departure time windows, durations and costs/revenues according to the aircraft type for each flight leg, find a fleet schedule that maximizes profits and satisfies certain additional constraints.*

A solution to this problem specifies the aircraft type and the departure time for each operational flight leg. It also provides the routing for each aircraft consisting of a sequence of flight legs and ground connections. Generally, when significant changes are made to flight departure times, there is an impact on the demand for those flights and consequently on the revenue. However, when flight departure time window width remains fairly narrow, it is realistic to assume that this will not impact demand. We make this flight independence assumption throughout the paper.

Some additional constraints that can be considered are: the number of available aircraft for each type, the restrictions on certain aircraft types at certain times and stations (e.g., curfews), the required thrus (i.e., connections between two flight legs imposed by the airline) and the limits on daily service at certain stations. The problem may also include the possibility of using shorter than normal connection times between two flight legs and the possibility of adding flight legs selected from a set of potential flight legs to daily schedules.

If the set of flight legs is balanced (i.e., at each station there are as many arrivals as there are departures of the same type), one can further impose at each station the availability of an equal number of aircraft of each type at the beginning and at the end of the day. However, because of time window width, these conditions do not ensure a periodic schedule, i.e., replicable every day. Consider for example, a flight leg that spans part of two consecutive days, i.e., it is in progress at the time junction of these two days. Even though at the end of the first day, the set of flight legs is balanced, this leg may still not be able to depart due to a deficit of aircraft. This is created by another leg which, while arriving in its time window, does so later than the departure time of the first leg. Since passenger demand varies daily and the flights offered depend on the day of week, we do not impose that the schedule be periodic. The periodic daily aircraft routing and scheduling problem is much more difficult as it will be seen from the brief discussion given in Section 6.

**Literature Review:** The literature on the DARSP and related problems dates back to the early 1960's. Most of the papers until the early 1980's have considered small airline problems or parts of the extended version of the problem such as determining the flights to be flown, or their frequency, or scheduling the aircraft using predetermined routes. Furthermore, most of these papers do not consider multiple aircraft types, departure time windows or the additional constraints of the DARSP mentioned above. They use heuristic solution approaches since computers were much less powerful at that time. The survey of Etschmaier and Mathaisel (1984) provides a good review of these papers.

[Levin \(1971\)](#) was the first to have proposed a model for the DARSP that included variable departure times for the flight legs. The departure times could only take the finite number of values obtained by discretizing the time windows of the corresponding flight legs. The problem was formulated as an integer linear program with bundle constraints. An interesting aspect of this formulation is that the linear relaxation resulted in an integer solution most of the time. However, this model does not consider multiple aircraft capacities, nor the additional constraints of the DARSP. Agin and Cullen (1975) have treated transportation routing and vehicle loading problems. They presented a mixed integer programming model, similar to a multi-commodity network flow model, that can be adapted to DARSP using a discretization of the time windows. It is solved using a heuristic based on Dantzig-Wolfe decomposition. For each aircraft, there is one subproblem that generates promising routes; when such a route is added to the master problem, the previous route of the corresponding aircraft is deleted. Hence, the integrality requirement that a single route be assigned to each aircraft is always satisfied.

Recently, Abara (1989) has proposed an integer linear programming approach to solve the DARSP with fixed flight departure times. The model is a multi-commodity network flow problem with additional constraints to limit, for example, the number of aircraft overnighing or the number of slots used at a station. The author does not specify the technique used to solve this large integer program, but reports good results for real-life problems at American Airlines.

Hane *et al.* (1993) have also presented a multi-commodity network flow model to solve the DARSP without departure time windows. Their model used an aggregation of the flow variables of Abara's model (see Section 4), and hence is adequate to determine the aircraft type assigned to each flight leg. However, the model requires human intervention for aircraft routing. Their approach consists of an interior point

LP algorithm, a cost perturbation strategy and a branch-and-bound procedure based on a generalized upper-bound dichotomy branching rule. To render this approach computationally tractable for large problems, the authors fix many flight connections *a priori*. This produces an overall heuristic method and substantially reduces the size of the problems considered. The resulting problems are comparable in size to those solved in this paper. The authors report successful computational experience using this approach.

**Overview and Contribution of the Paper:** We present two equivalent formulations for the DARSP. Both these integer programming models are solved by branch-and-bound. In the first model, we define a binary variable for each possible schedule for an aircraft type giving rise to a large Set Partitioning type problem (Section 2). To obtain lower bounds, we optimally solve its linear relaxation using a column generation approach. For each type of aircraft, there is one subproblem that generates promising schedules. The subproblem’s solution is obtained by solving a longest path problem with time windows. We describe the underlying time-space network in detail in Section 3.

In the second model, a binary variable represents a possible connection between two flight legs performed by a particular aircraft. For each aircraft type, we define these variables on a time-space network identical to the one used in the column generation solution approach of the first model. This second formulation gives rise to a multi-commodity network flow model with the usual flow variables and also time variables (Section 4). Abara (1989) and to some extent, Hane *et al.* (1993) have developed special cases of this generalized model. Our model can include all constraints considered to date in addition to flexible departure times.

The second formulation also provides the fundamental concepts necessary to develop optimal branching strategies compatible with the column generation solution approach used for the first model (Section 5). This mechanism is not well known in the literature as noted, for example, by Chvátal (1983, pp.197-198) in the context of the Cutting-Stock problem. As shown in the survey by Desrosiers *et al.* (1994), multi-commodity network flow type models can also be used for many other fixed or variable time constrained vehicle routing and crew scheduling problems. In this regard, we consider some important extensions of the DARSP model in Section 6. Finally, Section 7 presents the computational results we have obtained on real-world test problems and Section 8 our conclusions.



## 2 A Set Partitioning Type Formulation

In this section, we formulate the DARSP as a Set Partitioning problem with additional constraints. We also propose a solution strategy to solve the linear relaxation of this formulation.

**Notation:** Let  $N$  be the set of operational flight legs and  $K$  the set of aircraft types. Denote by  $n^k$  the number of available aircraft of type  $k \in K$ . Define  $\Omega^k$ , indexed by  $p$ , as the set of feasible schedules for aircraft of type  $k \in K$  and let index  $p = 0$  denote the empty schedule for an aircraft. Next associate with each schedule  $p \in \Omega^k$  the value  $c_p^k$  denoting the anticipated profit if this schedule is assigned to an aircraft of type  $k \in K$  and  $a_{ip}^k$  a binary constant equal to 1 if this schedule covers flight leg  $i \in N$  and 0 otherwise. Furthermore, let  $S$  be the set of stations and  $S^k \subseteq S$  the subset having the facilities to serve aircraft of type  $k \in K$ . Then, define  $o_{sp}^k$  and  $d_{sp}^k$  to equal to 1 if schedule  $p$ ,  $p \in \Omega^k$ , starts and ends respectively at station  $s$ ,  $s \in S^k$ , and 0 otherwise.

Denote by  $\theta_p^k$ ,  $p \in \Omega^k \setminus \{0\}$ ,  $k \in K$ , the binary decision variable which takes the value 1 if schedule  $p$  is assigned to an aircraft of type  $k$ , and 0 otherwise. Finally, let  $\theta_0^k$ ,  $k \in K$ , be a non-negative integer variable which gives the number of unused aircraft of type  $k$ .

**Formulation:** Using these definitions, the DARSP can be formulated as:

$$\text{Maximize } \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \theta_p^k \quad (1)$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k \theta_p^k = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{p \in \Omega^k} (d_{sp}^k - o_{sp}^k) \theta_p^k = 0, \quad \forall k \in K, \forall s \in S^k \quad (3)$$

$$\sum_{p \in \Omega^k} \theta_p^k = n^k, \quad \forall k \in K \quad (4)$$

$$\theta_p^k \geq 0, \quad \forall k \in K, \forall p \in \Omega^k \quad (5)$$

$$\theta_p^k \text{ integer}, \quad \forall k \in K, \forall p \in \Omega^k. \quad (6)$$

The objective function (1) states that we wish to maximize the total anticipated profit. Constraints (2) require that each operational flight leg be covered exactly once. Constraints (3) correspond to the flow conservation constraints at the beginning and the end of the day at each station and for each aircraft type. Constraints (4) limit the number of aircraft of type  $k \in K$  that can be used to the number available. Finally, constraints (5) and (6) state that the decision variables are non-negative integers. This model is a Set Partitioning problem with additional constraints.

**Solution Strategy:** We have chosen a branch-and-bound approach. Branching strategies will be explored in detail in Section 5. Lower bounds are obtained by utilizing a column generation technique to solve the linear relaxation (1)-(5) of this problem (see Lasdon 1970). The technique consists of dividing a linear program into a restricted master problem and a subproblem. The role of the restricted master problem, consisting in the present case of the linear program (1)-(5) defined over a relatively small number of feasible schedules, is to find a current optimal solution and to compute the dual variables associated with this current solution. It can be solved using the simplex algorithm.

On the other hand, the subproblem is used to test whether the current solution is optimal over all possible schedules, i.e., over  $\bigcup_{k \in K} \Omega^k$ . The subproblem is a longest path problem with time windows which uses the dual variables of the current optimal solution of the master problem. If the current solution is not optimal, the subproblem must provide a new feasible schedule to be added to the restricted master problem. The subproblem can be solved using dynamic programming ([Desrosiers \*et al.\* 1983](#), [Desrochers and Soumis 1988a,b](#)).

For additional applications of the column generation technique to various vehicle routing and crew scheduling problems, the reader is referred to the survey paper by [Desrosiers \*et al.\* \(1994\)](#).

### 3 The Subproblem Network Structure

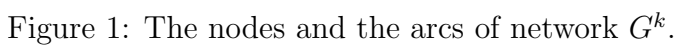
A specific network  $G^k = (V^k, A^k)$ , where  $V^k$  is the set of nodes and  $A^k$  is the set of arcs, is used for each aircraft type  $k \in K$  to generate feasible schedules. The same

networks will also be used later for the time constrained multi-commodity network flow model. Each network contains five node types and seven arc types as illustrated in Figure 1.

**The Nodes:** The five node types in  $V^k$  are: *source*, *sink*, *initial station*, *final station* and *flight*. There is a single source node  $o(k)$  and a single sink node  $d(k)$ . These two nodes represent the start and the end, respectively, of all paths corresponding to the schedules for aircraft type  $k \in K$ . An initial station node  $s \in S^k$  and a final station node  $t \in S^k$  indicate the stations where a schedule starts and ends, respectively. The sets of initial and final station nodes are denoted by  $S_1^k$  and  $S_2^k$ , respectively. Remark that these two sets contain the same stations due to the flow conservation constraints for each aircraft type at each station. Finally, a flight node  $i$  ( $i \in N^k \subseteq N$ ) simply represents an operational flight leg to be flown. There can be several nodes representing the same flight leg in different networks when this leg can be flown by different types of aircraft. To summarize, the set of nodes in  $G^k$  is given by:

$$V^k = N^k \cup S_1^k \cup S_2^k \cup \{o(k), d(k)\}.$$

**The Arcs:** The seven arc types in  $A^k$  are: *empty*, *source*, *sink*, *schedule start*, *schedule end*, *turn* and *short turn*. There is a single empty arc  $(o(k), d(k))$  which represents an empty aircraft schedule. This arc forms the set  $OD^k$ . Note that this empty arc  $(o(k), d(k))$  can be removed from  $A^k$  since the corresponding empty schedule variable  $\theta_0^k$  can be introduced with cost  $c_0^k$  in the restricted master problem prior to the column generation process. A source arc  $(o(k), s)$ ,  $s \in S_1^k$ , part of the arc set denoted by  $OS_1^K$ , goes from the source node to an initial station node. A sink arc  $(s, d(k))$ ,  $s \in S_2^k$ , belonging to the arc set  $S_2D^k$ , links a final station node to the sink node. A schedule start arc  $(s, j)$ ,  $s \in S_1^k$ ,  $j \in N^k$ , part of the arc set  $S_1N^k$ , begins at an initial station node and ends at a flight node whose origin is station  $s$ . Similarly, a schedule end arc  $(i, s)$ ,  $i \in N^k$ ,  $s \in S_2^k$ , belonging to the arc set  $NS_2^k$ , goes from a flight node to the final station node where this flight leg ends. A turn arc  $(i, j)$ ,  $i$  and  $j \in N^k$ , included in the arc set  $NN^k$ , links two flight nodes and simply represents a normal connection between these two flight legs. Such an arc exists between two flight nodes if the flight legs can be flown consecutively by the same aircraft while respecting the time windows of each flight leg and the normal connection time of the aircraft at the connecting station. Formally, let  $a_i^k$  be the earliest time at which leg  $i$  can start,  $b_j^k$



the latest time at which leg  $j$  can start,  $l_i^k$  the duration of leg  $i$ , and  $t_{ij}^k$  the normal connection time between legs  $i$  and  $j$ . Then, there exists a turn arc between flight nodes  $i$  and  $j$  in  $G^k$  if  $a_i^k + l_i^k + t_{ij}^k \leq b_j^k$ , and the arrival station of leg  $i$  is the same as the departure station of leg  $j$ . Finally, short (or quick) turn arcs, belonging to the arc set denoted by  $NQN^k$ , are turn arcs involving a connection time  $s_{ij}^k$  that is shorter than normal, i.e.,  $s_{ij}^k < t_{ij}^k$ . Such an arc exists if  $a_i^k + l_i^k + s_{ij}^k \leq b_j^k$ . This may be the only arc between nodes  $i$  and  $j$ , or a turn arc may also exist. Note that if  $b_i^k + l_i^k + t_{ij}^k \leq a_j^k$ , there is no need to include a short turn arc because a normal connection always fits. Since normal connection times are preferable to short connection times, short turn arcs incur penalty costs. To conclude, the set of arcs in  $G^k$  is given by:

$$A^k = OD^k \cup OS_1^k \cup S_1N^k \cup NN^k \cup NQN^k \cup NS_2^k \cup S_2D^k.$$

A schedule generated by the subproblem must meet two requirements: it must be feasible and its marginal profit must be positive. To test whether these requirements are satisfied, information is kept on each arc concerning the duration of the activities on the arc and the costs/revenues related to these activities.

**Schedule Feasibility:** The feasibility of a schedule is tested in the longest path algorithm using constrained time variables representing the departure times of the flight legs. To be feasible, schedules must satisfy time window constraints concerning the departure times. That is, for each flight leg  $j$ , the value of the corresponding time variable must be within the time window  $[a_j^k, b_j^k]$ . Hence, to compute the values of the time variables and test schedule feasibility, we are required to know the duration of the activities (flight legs and connections) on each arc of the network. We have chosen to place the duration of a flight leg on the arc leaving the flight node. Table 1 summarizes the duration of each arc type and the additional information described below.

**The Cost/Revenue Structure:** The longest path algorithm finds a feasible schedule with the largest marginal profit. The marginal profit is calculated using the revenue and cost of each activity and the dual variables associated with the current optimal solution of the restricted master problem. The anticipated cost of an empty schedule for a type  $k$  aircraft is  $e_o^k$ , so the profit coefficient on arc  $(o(k), d(k))$  is  $-e_o^k$ ,  $k \in K$ . Next, the profit of flight leg  $j$  is placed on every arc leaving flight node  $j$ .

ARCS	TYPE	SET	DURATION	PROFIT	DUAL VARIABLES
$(o(k), d(k))$	empty	$OD^k$	0	$-e_o^k$	$-\beta^k$
$(o(k), s), s \in S_1^k$	source	$OS_1^k$	0	$-M$ or 0	$-\beta^k$
$(s, j), s \in S_1^k, j \in N^k$	schedule start	$S_1N^k$	0	0	$\sigma_s^k$
$(i, j), i$ and $j \in N^k$	turn	$NN^k$	$l_i^k + t_{ij}^k$	$r_i^k - e_i^k$	$-\alpha_i$
$(i, j), i$ and $j \in N^k$	short turn	$NQN^k$	$l_i^k + s_{ij}^k$	$r_i^k - e_i^k - q_{ij}^k$	$-\alpha_i$
$(i, s), i \in N^k, s \in S_2^k$	schedule end	$NS_2^k$	$l_i^k$	$r_i^k - e_i^k$	$-\alpha_i - \sigma_s^k$
$(s, d(k)), s \in S_2^k$	sink	$S_2D^k$	0	0	0

Table 1: The information associated with each arc in  $A^k$

This profit is equal to  $r_i^k - e_i^k$ , where  $r_i^k$  and  $e_i^k$  are the anticipated revenue and cost, respectively, if flight leg  $i \in N^k$  is assigned to an aircraft of type  $k \in K$ . To favor normal connections between flight legs, a penalty  $q_{ij}^k$  is imposed on the arc between legs  $i$  and  $j$  in  $G^k$  if this is a short turn arc. Finally, to minimize the number of aircraft required to cover all operational flight legs, a large fixed cost  $M$  is placed on every source arc. For a fixed fleet, the source arcs are assigned a zero profit.

In addition to revenues and costs, the dual variables must be placed on the appropriate arcs in the network. Let  $\alpha_i$ ,  $\sigma_s^k$ , and  $\beta^k$ ,  $\forall i \in N$ ,  $\forall k \in K$ , and  $\forall s \in S^k$ , be the dual variables associated with constraints (2), (3) and (4), respectively. The marginal profit  $\bar{c}_p^k$  of a schedule  $p \in \Omega^k$  is calculated as:

$$\bar{c}_p^k = c_p^k - \sum_{i \in N^k} \alpha_i a_{ip}^k - \sum_{s \in S^k} \sigma_s^k (d_{sp}^k - o_{sp}^k) - \beta^k, \quad (7)$$

where  $c_p^k$ , the anticipated profit of schedule  $p$ , is equal to the sum of the profits on the arcs forming the path corresponding to schedule  $p$ . Consequently, to correctly compute the marginal profit of a schedule  $p \in \Omega^k$  using a longest path algorithm, the following values must be assigned to the appropriate arcs of  $G^k$ :  $-\alpha_i$ ,  $i \in N^k$ , on all arcs leaving flight node  $i$ ,  $-\beta^k$  on all source arcs and on the empty arc,  $-\sigma_s^k$ ,  $s \in S_2^k$ , on all final station arcs entering the final station node  $s$ , and  $\sigma_s^k$ ,  $s \in S_1^k$ , on all initial station arcs leaving the initial station node  $s$ .

**Specialized Representations:** The definition of a flight leg can be extended to include sequences of consecutive operational flight legs that must be assigned to the same aircraft as required by the airline. Such a sequence is called a thru and only

one covering constraint (2) is necessary in the Set Partitioning type formulation of Section 2 to account for all operational flight legs in it. Similarly, if such a thru can be assigned to an aircraft of type  $k \in K$ , then only one flight node is necessary to represent it in  $G^k$ .

In this paper, we assume that the networks  $G^k$ , for all  $k \in K$ , are acyclic since time windows of relatively small width are considered. If this was not the case, any  $G^k$  could be transformed into an acyclic network by dividing the time windows of the nodes included in cycles and creating a distinct node copy for each subdivision. This is equivalent to having several time windows for the service at customers in the time constrained vehicle routing problem (Desrochers, Desrosiers and Solomon 1992).

Other network representations such as time-line networks described in [Barnhart \*et al.\* \(1991\)](#) for crew scheduling problems and in [Hane \*et al.\* \(1993\)](#) for aircraft routing problems can be equivalently used. The network defined in this section was preferred over the others because it simplifies the presentation of the following multi-commodity network flow formulation.

## 4 A Time Constrained Multi-Commodity Network Flow Formulation

This section formulates the DARSP as a time constrained multi-commodity network flow formulation. First we introduce the notation which is followed by the formulation. Next we discuss a solution strategy to solve it. In the next section, we will refer to this formulation to design branching strategies for the Set Partitioning type formulation (1)-(6).

**Notation:** First, let  $X_{ij}^k$  be the integer flow variable corresponding to type  $k$  aircraft,  $k \in K$ , on the arc  $(i, j) \in A^k$ . This variable indicates the number of type  $k$  aircraft using arc  $(i, j)$ . Next, let  $T_i^k$ ,  $k \in K$ , be a time variable defined at node  $i \in V^k$ . If  $i \in N^k$ , this variable is the departure time of flight leg  $i$  within the time interval  $[a_i^k, b_i^k]$ . Otherwise, it is fixed at 0 for all source and initial station nodes and corresponds to the latest arrival time at all final station and sink nodes. Denote by  $d_{ij}^k$  the duration of the activities on arc  $(i, j) \in A^k$  depending on the aircraft type  $k \in K$ . These arc durations have already been presented in Table 1.

**Formulation:** The proposed time-constrained multi-commodity network flow formulation is given by:

$$\text{Maximize } \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij}^k X_{ij}^k \quad (8)$$

subject to:

$$\sum_{k \in K} \sum_{j: (i,j) \in A^k} X_{ij}^k = 1, \quad \forall i \in N \quad (9)$$

$$\sum_{i: (i,s) \in NS_2^k} X_{is}^k - \sum_{j: (s,j) \in S_1 N^k} X_{sj}^k = 0, \quad \forall k \in K, \forall s \in S^k \quad (10)$$

$$\sum_{s \in S_1^k} X_{o(k),s}^k + X_{o(k),d(k)}^k = n^k, \quad \forall k \in K \quad (11)$$

$$\sum_{i: (i,j) \in A^k} X_{ij}^k - \sum_{i: (j,i) \in A^k} X_{ji}^k = 0, \quad \forall k \in K, \forall j \in V^k \setminus \{o(k), d(k)\} \quad (12)$$

$$\sum_{s \in S_2^k} X_{s,d(k)}^k + X_{o(k),d(k)}^k = n^k, \quad \forall k \in K \quad (13)$$

$$X_{ij}^k \geq 0, \quad \forall k \in K, \forall (i,j) \in A^k \quad (14)$$

$$a_i^k \leq T_i^k \leq b_i^k, \quad \forall k \in K, \forall i \in V^k \quad (15)$$

$$X_{ij}^k (T_i^k + d_{ij}^k - T_j^k) \leq 0, \quad \forall k \in K, \forall (i,j) \in A^k \quad (16)$$

$$X_{ij}^k \text{ integer}, \quad \forall k \in K, \forall (i,j) \in A^k. \quad (17)$$

The objective function (8) maximizes the total profit made from daily schedules. Relations (9) require that each operational flight leg be covered exactly once. Relations (10) represent the flow conservation constraints for each aircraft type at each station, i.e., the number of aircraft of type  $k \in K$  at station  $s \in S^k$  must be the same at the beginning and at the end of a day. Relations (11) ensure that a maximum of  $n^k$  schedules be assigned to aircraft of type  $k \in K$ . Relations (12) are the usual network flow conservation constraints. Given (11) and (12), relations (13) are redundant. Together, these constraints describe a path structure with  $n^k$  units flowing from  $o(k)$  to  $d(k)$ . Relations (14) are the nonnegativity constraints. Relations (15) define the time window constraints. Relations (16) ensure the compatibility between the flow and the time variables. Finally, integrality constraints on the flow variables are given by relations (17).

Removing the coupling constraint set (10) and adding capacity constraints, the above model becomes the vehicle routing problem with time windows. Two successful solution approaches have been described for this problem: Desrochers, Desrosiers and



Solomon (1992) have proposed a column generation based scheme while Kohl and Madsen (1993) have used a Lagrangean Relaxation method.

**Solution Strategy:** The previous time constrained multi-commodity network flow model can be solved using a decomposition technique such as Dantzig-Wolfe or Lagrangean relaxation embedded in a branch-and-bound search tree. Using the former, formulation (8)-(17) is divided into a master problem, consisting of the objective function (8) and the constraint sets (9) and (10), and a subproblem which considers the relations (11)-(17) and a modified objective function to be described later. The master problem retains the flight covering constraints and the flow conservation constraints for each type of aircraft at each station. The subproblem, which is separable by aircraft type  $k$ , is a longest path problem with time windows and a supply of  $n^k$  units at the source node of  $G^k$ .

**Models for Fixed Time-Tabled Flights:** By removing relations (15) and (16) which concern the time dimension, the above model becomes an Abara (1989) type model. On the other hand, by defining additional variables as:

$$X_i^k = \sum_{j:(i,j) \in A^k} X_{ij}^k, \quad \forall k \in K, \forall i \in N^k,$$

one can recognize the structure of the model of Hane *et al.* (1993). The new variables determine if the aircraft type  $k$  is assigned to flight leg  $i$ . Their objective function can be obtained using these new variables if short connection times are not considered in our model. In fact, the cost coefficients in (8) can be rewritten as  $c_i^k = r_i^k - e_i^k$ ,  $k \in K$ ,  $i \in V^k$ , instead of  $c_{ij}^k$ ,  $k \in K$ ,  $(i, j) \in A^k$  when short connection times are forbidden. The flight covering constraints can be obtained directly. The other constraints of their formulation (availability of aircraft and flow conservation) are not identical to ours as they defined the flow variables on a time-line network and also considered periodic schedules.

## 5 Branch-and-Bound Strategies

In the time constrained multi-commodity network flow formulation (8)-(17), the branching decisions used to obtain integer solutions must naturally be taken on the

flow or the time variables. This is also the case if a Lagrangean relaxation scheme is applied to (8)-(17).

On the other hand, if the Set Partitioning type model (1)-(6) is used, the path variables  $\theta_p^k$ ,  $p \in \Omega^k$ ,  $k \in K$ , may be considered for branching decisions. A fractional variable  $\theta_p^k$ ,  $0 < \theta_p^k < 1$ , can easily be set to 1: it consists of deleting all the flight legs covered by the corresponding path in all networks  $G^k$ ,  $k \in K$ , as well as their corresponding flight covering constraints in the master problem, and in adjusting the other constraints accordingly. On the other hand, setting  $\theta_p^k$  to 0 is useless since its corresponding path will be generated again in the optimal solution of the linear relaxation. Therefore the branching decisions cannot be taken directly on the path variables. However there exist several ways to transfer the effects of these decisions to the subproblem variables. For example, setting  $\theta_p^k = 0$  is equivalent to fixing at 0 one or many of the flow variables on the corresponding path, leading to several branches in the branching tree (Desrosiers *et al.* 1984).

Alternatively, knowing the values of the path variables, one can evaluate the network flow variables. Indeed, for single- and multi-commodity network flow problems, it is well known that the integer node-arc and node-path formulations are equivalent (see Ahuja, Magnanti and Orlin 1993). However it is much less known that the node-path formulation can be obtained from the node-arc formulation by applying a Dantzig-Wolfe decomposition, even if such a result dates back to the sixties (see Tomlin 1965, and Jarvis 1969). In our case, the time constrained multi-commodity network flow formulation is decomposed as follows: the constraints (9)-(10) involving several commodities are kept in the master problem, while the local path constraints (11)-(17) involving a single commodity appear in the subproblems.

Therefore, even if the Set Partitioning type model is used, optimal branching decisions can be taken on the  $X_{ij}^k$  variables restricted to integer values. These flow variables appear in both the subproblem structure of the column generation scheme and the multi-commodity flow model. This fundamental observation allows to develop branching strategies compatible with the column generation technique.

In addition to branching decisions taken on the flow variables, one can take binary decisions on linear combinations of them such as

$$X_{ij} = \sum_{k \in K} X_{ij}^k, \quad i, j \in N^k, \text{ and} \quad (18)$$

$$X_i^k = \sum_{j: (i,j) \in A^k} X_{ij}^k, \quad k \in K, i \in N^k. \quad (19)$$

A decision to set  $X_{ij}$  at 1 in (18) imposes a connection between flight legs  $i$  and  $j$  whatever the aircraft type. Setting  $X_i^k = 1$  in (19) forces the assignment of aircraft type  $k$  to flight leg  $i$ .

Another useful set of variables is

$$X_{o(k)}^k = \sum_{s \in S_1^k} X_{o(k),s}^k, \quad k \in K \quad (20)$$

where each variable counts the number of aircraft of type  $k$  used in the current solution. This number is restricted to take integer values and hence any branching decision can easily be incorporated by defining two-side inequalities in (4) or (11).

Other branching strategies based on the variables of the multi-commodity network flow model can also be conceived. Not only flow variables can be used in numerous ways, but also time variables can lead to several appealing branching rules (Gélinas *et al.* 1995).

## 6 Extensions

In this section, we discuss how to modify the two formulations presented in Sections 2 and 4 to account for several interesting extensions of the DARSP.

**Potential Flight Legs:** Situations arise where in addition to  $N$ , the set of flight legs to cover, there exists a set of potential flight legs from which it is possible to select additional flight legs to be flown. Denote by  $P$  this set of potential flight legs which contains different subsets,  $P^k \subseteq P$ ,  $k \in K$ , for each type of aircraft. For each potential flight leg in  $P^k$ ,  $k \in K$ , create a new potential flight node in  $G^k = (V^k, A^k)$  where

$$V^k = N^k \cup P^k \cup S_1^k \cup S_2^k \cup \{o(k), d(k)\},$$

and link it to the other nodes with schedule start, schedule end, turn and short turn arcs so that  $A^k$  is appropriately defined. To ensure that these potential flight legs are not covered more than once, the only new constraints that must be introduced are:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k \theta_p^k \leq 1, \quad \forall i \in P, \text{ in model (1)-(6), and}$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A^k} X_{ij}^k \leq 1, \quad \forall i \in P, \text{ in model (8)-(17).}$$

**Dependent Flight Legs:** The problem treated here can be extended to consider dependent flight legs. This would require the solution of a stochastic nonlinear multi-commodity integer program over the probability distribution of flight demands defined on the the flights' departure time windows. An approximate solution to such a clearly nontractable program could be obtained by partitioning the departure time windows into a certain number of very narrow time intervals. We assume that demand remains constant within each of the time intervals of the partition. For each node, we define as many copies of it as there are time windows in the partition. Constraint set (2) or (9) will ensure that the model will choose exactly one among these.

**Periodic Daily Schedule for Fixed Time-Tabled Flights:** Given a set of flight legs balanced over a daily horizon, i.e., for each station, there are as many departures as there are arrivals, there could be a need to construct periodic daily schedules to be repeated day after day. This periodic daily schedule problem without time windows has been solved by Abara (1989) and by Hane *et al.* (1993).

Consider a horizon of two consecutive days. Since the daily schedule is replicable day after day, the deployment of the airline fleet with respect to aircraft types must be identical each of the two days. That is, if  $t$  is a given time of the first day and  $D$  a 24-hour duration constant, then at time  $t$  and  $t + D$  there must be the same number of aircraft of each type at each station, and the aircraft types assigned to the flight legs in progress must also be the same. From this observation, we infer that the specific time where we start the schedule might be selected arbitrarily. Let  $t_0$  be this time. If no flight legs are in progress at time  $t_0$  then the models presented in Sections 2 and 4 without time constraints are suitable for the periodic daily schedule problem.

On the other hand, if there are some flight legs in progress at time  $t_0$ , then the networks  $G^k$  must be modified. Observe that these legs are divided in two parts, one after  $t_0$  and the other before  $t_0$ . The aircraft assigned to the first part is not necessarily the same as the one assigned to the second; however, they must be of the same aircraft type. All aircraft assigned to the legs in progress must be taken into account as they might be assigned to subsequent flight legs.

Therefore, instead of associating a single flight node with a leg in progress, we associate two. The first node corresponds to the second part of the leg which occurs after  $t_0$  while the second node represents the first part which takes place before  $t_0 + D$ . Let  $N_1^k$  and  $N_2^k$  be the sets of first and second nodes associated with legs in progress,

respectively. The nodes in  $N_1^k$  and  $N_2^k$  are linked to the other nodes of  $G^k$  in the same manner as the initial station and the final station nodes, respectively. For each leg in progress two additional arcs must be added. One between the node in  $N_1^k$  corresponding to this leg and the final station node in  $S_2^k$  where this leg ends. The other connects the initial station node in  $S_1^k$  where this leg begins with the node in  $N_2^k$  associated with this leg. These additional arcs allow us to consider schedules containing only a single leg in progress.

In the formulations presented in Sections 2 and 4, the nodes in  $N_1^k$  and  $N_2^k$  can be included either in  $N^k$  or in  $S_1^k$  and  $S_2^k$ , respectively. If they are included in  $N^k$  (that is,  $N^k := N^k \cup N_1^k \cup N_2^k$ ), then equations (2) or (9) ensure the covering of the two parts of the flight legs in progress. However, an additional flow conservation constraint for each aircraft type and for each pair of corresponding nodes in  $N_1^k$  and  $N_2^k$  must be added to the two formulations to ensure that these nodes are covered by the same aircraft type. On the other hand, if the nodes are included in  $S_1^k$  and  $S_2^k$  (that is,  $S_1^k := S_1^k \cup N_1^k$  and  $S_2^k := S_2^k \cup N_2^k$ ), then covering constraints for the flight legs in progress must be added to the two models. In this case, equations (3) or (10) ensure that the two parts of a flight leg in progress are covered by the same aircraft type. Note that in both cases only one covering constraint is necessary for each pair of corresponding nodes in  $N_1^k$  and  $N_2^k$ , since the flow conservation constraints ensure that if one of them is covered then the other will be as well.

**Periodic Daily Schedule for Flexible Time-Tabled Flights:** In addition to the above modifications needed for the periodic daily schedule problem for fixed time-tabled flights, the flexible version may require additional constraints to ensure that the departure time of a flight leg in progress before  $t_0$  is the same as the departure time of this leg the next day. If no flight legs are in progress at time  $t_0$ , then the models presented in Sections 2 and 4 with time constraints are suitable for this flexible periodic daily schedule problem.

On the other hand, if there are some flight legs in progress at time  $t_0$ , let  $[a_i^k, b_i^k]$  be the time window of flight leg  $i \in N_2^k$ . Define then  $[a_i^k - D, b_i^k - D]$  as the time window on the corresponding leg  $i \in N_1^k$ . The time windows on the source node  $o(k)$ , the initial station nodes in  $S_1^k$ , the final station nodes in  $S_2^k$  and the sink node  $d(k)$  are replaced by the value  $a^k = \min_{i \in N_1^k} a_i^k$ . They become  $[a^k, a^k]$ ,  $[a^k, a^k]$ ,  $[a^k, \infty]$  and  $[a^k, \infty]$ , respectively.

Furthermore, the time window constraints (15) for flight legs  $i \in N_1^k \cup N_2^k$  are modified to

$$a_i^k \left( \sum_{j:(i,j) \in A^k} X_{ij}^k \right) \leq T_i^k \leq b_i^k \left( \sum_{j:(i,j) \in A^k} X_{ij}^k \right), \quad \forall k \in K, \forall i \in N_1^k \cup N_2^k.$$

In the optimal solution, if  $i \in N_1^k \cup N_2^k$ ,  $k \in K$ , then due to constraints (9), (14) and (17),  $\sum_{j:(i,j) \in A^k} X_{ij}^k$  is restricted to be binary, i.e., it is equal to 1 if an aircraft of type  $k$  visits node  $i$ , and 0 otherwise. Therefore, the time variables belong to  $[a_i^k, b_i^k]$  if node  $i$  is covered, and equal 0 otherwise.

Denote by  $(u, u') \in N_1^k \times N_2^k$  the pair of nodes corresponding to a flight leg in progress performed by an aircraft of type  $k$ . The compatibility of the departure times of this leg in progress is then ensured by the constraint:

$$T_u^k + D = T_{u'}^k, \quad (u, u') \in N_1^k \times N_2^k,$$

which can also be written as:

$$T_u^k - a_u^k = T_{u'}^k - a_{u'}^k, \quad (u, u') \in N_1^k \times N_2^k.$$

Since the aircraft type which will cover this leg is unknown and the value of  $T_u^k - a_u^k \left( \sum_{j:(u,j) \in A^k} X_{uj}^k \right)$  is equal to 0 if leg  $u \in N_1^k \cup N_2^k$  is not covered by an aircraft of type  $k \in K$ , the equations ensuring the compatibility of the departure times of the legs in progress are:

$$\sum_{k \in K} (T_u^k - a_u^k \left( \sum_{j:(u,j) \in A^k} X_{uj}^k \right)) = \sum_{k \in K} (T_{u'}^k - a_{u'}^k \left( \sum_{j:(u',j) \in A^k} X_{u'j}^k \right)), \quad \forall (u, u') \in N_1^k \times N_2^k. \quad (21)$$

Constraints (21) are coupling constraints and hence cannot be considered in the subproblem. Note that these coupling constraints include both time and flow variables. However, if  $a_u^k = a_u$ ,  $\forall k \in K$  and  $\forall u \in N_1^k \cup N_2^k$ , then the flow variables can be removed from (21) since  $\sum_{k \in K} \sum_{j:(u,j) \in A^k} X_{uj}^k = 1$  by the covering constraints (9).

One can observe that the subproblem retains only the local time constraints, i.e., the time window on a single path, while the coupling constraints (21) permit us to link the time variables on two different paths. When these coupling constraints are relaxed in a Lagrangean relaxation (or kept in the restricted master problem of a Dantzig-Wolfe decomposition), the time variables appear in the objective function

together with dual multipliers. In this case, the marginal profit of a new schedule varies with the marginal cost of the arcs and also with the visit times of nodes in  $N_1^k \cup N_2^k$ . This problem is much more difficult and requires a different time constrained optimization path algorithm ([Ioachim \*et al.\* 1994](#)).

## 7 Computational Results

All the test problems presented in this section were solved using the GENCOL software, developed at the GERAD research center, which relies on the optimizer CPLEX (CPLEX 1993) to solve linear programs. This software package consists essentially of a branch-and-bound algorithm where the upper bound at each node is obtained by solving the linear relaxation (1)-(5) of the Set Partitioning type formulation using the column generation technique described in Section 2.

**Solution Strategies:** The branching decisions were taken on the following linear combination of the flow variables:  $X_{ij} = \sum_{k \in K} X_{ij}^k$ , where  $i$  and  $j \in N$ . Fixing the binary variable  $X_{ij}$  at 1 corresponds to imposing a connection between flight legs  $i$  and  $j$ . A depth first strategy was used. Although optimal solutions can be obtained through an exhaustive exploration of the branch-and-bound search tree, a heuristic procedure has been used to limit the computational time. The following rules were applied: 1) stop prematurely the solution process of the restricted master problem at a branching node if there is no improvement over the last 10 iterations; 2) fix at 1 the path variables that have a value very close to 1, i.e., greater than or equal to 0.95; and 3) fathom a pending node if the difference between its upper bound and the best integer solution found so far is less than \$200.

Two series of test problems were solved by GENCOL. The information concerning the size of these problems is summarized in Table 2.

**North American Carrier Test Problems:** The first set of data corresponds to a typical weekday for a medium haul fleet of a North American airline. There are 204 flow conservation constraints for each aircraft type at each station in addition to the 383 flight covering constraints of the master problem. The objective is first to minimize the number of aircraft utilized to cover all the flight legs and then to maximize the anticipated profit. Four scenarios, differing in terms of the flexibility

	North American Carrier	European Carrier
Number of Flight Legs	383	252
Number of Aircraft	91	51
Number of Types	9	6
Number of Stations	33	44
Short Turn Arcs	yes	no

Table 2: Size information for the test problems.

of the flight departure times, were solved using GENCOL. The first considered fixed departure times while the other three considered departure time windows (centered at the fixed departure times of the first scenario) of width  $\pm 10$ ,  $\pm 20$  and  $\pm 30$  minutes. The  $\pm 30$  minutes scenario is presented primarily for numerical interest. One would expect that in practice, time windows of such width would begin to affect demand and therefore the objective function.

The airline who supplied these data also provided a solution for the fixed departure time case utilizing the 91 aircraft. However comparisons between their solution and the solutions obtained by GENCOL cannot be made since they also considered a cargo transportation component which was not described in the data. Nevertheless the following results show the usefulness and the efficiency of the column generation approach in this case.

Table 3 presents the results obtained by GENCOL for the four scenarios. Each column corresponds to a scenario. The rows of the table are divided in four parts. The first part specifies the size of the subproblem time-line network. The second part gives information about the solution process of the linear relaxation problem. It indicates the value  $z_{LP}^H$  of the heuristic solution (which has been verified, for each of these four scenarios, to be equal to the value  $z_{LP}^*$  of the optimal solution), the number of aircraft utilized in this solution, the number of iterations (i.e., the number of calls to the restricted master problem) performed to obtain this solution, the CPU times to solve the subproblem and the restricted master problem as well as the total CPU time on a HP 9000/735 workstation (86.7 Specfp92, 52.0 Specint92). The third part gives the value of the heuristic integer solution ( $z_{IP}^H$ ) and the numbers of aircraft and short time connections utilized in this solution. It also specifies the number of nodes explored in the branch-and-bound search tree. Finally, in addition to the



	$\pm 0$ MIN	$\pm 10$ MIN	$\pm 20$ MIN	$\pm 30$ MIN
<b>Network Information:</b>				
Number of Nodes	2070	2118	2617	2765
Number of Arcs	4070	4345	5500	6498
<b>Linear Relaxation:</b>				
$z_{LP}^H$ ( $= z_{LP}^*$ )	\$ 30 141	\$ 30 204	\$ 29 335	\$ 27 183
Number of Aircraft	79	76	72	68
Number of Iterations	77	95	127	135
$CPU_{SP}$	90	109	59	75
$CPU_{MP}$	106	131	246	292
$CPU_{TOT}$	196	240	305	367
<b>Integer Solution:</b>				
$z_{IP}^H$	\$ 30 141	\$ 30 204	\$ 29 252	\$ 27 047
Number of Aircraft	79	76	72	68
Number of Short Turns	11	8	14	34
Number of B&B Nodes	2	0	8	14
<b>Global Information:</b>				
Number of Iterations	177	95	380	708
Number of Columns	20 644	15 467	72 238	171 050
$CPU_{SP}$	195	109	367	947
$CPU_{MP}$	234	131	1102	2561
$CPU_{TOT}$	429	240	1469	3508
Absolute Integrality Gap	\$ 0	\$ 0	\$ 83	\$ 136
Relative Integrality Gap	0 %	0 %	0.28 %	0.5 %

Table 3: GENCOL's results on the North American carrier data.

overall CPU times, the global information part indicates the numbers of iterations and columns generated during the overall solution process, as well as the absolute and relative integrality gaps between the optimal linear relaxation solution and the heuristic integer solution, i.e.,  $z_{LP}^* - z_{IP}^H$  and  $(z_{LP}^* - z_{IP}^H)/z_{LP}^*$ , respectively.

Observe that the number of nodes and arcs of the subproblem network increases with the width of the time windows. This comes from the duplication of certain flight nodes which is required whenever a flight leg can be connected to a subsequent leg if the departure time of the first leg is early enough, but cannot be connected to that same leg if this departure time occurs too late in the time window.

The linear relaxation part shows that the anticipated profit and the number of aircraft utilized decrease as the time windows become wider. The decrease in the number of aircraft is due to the fact that more schedules are feasible as the flexibility of the departure times is increased. Since the main objective is to minimize the number of aircraft utilized, a larger set of feasible schedules leads to a reduction in the number of aircraft utilized. Compared to the scenario without time window flexibility which requires 79 aircraft, the  $\pm 10$ ,  $\pm 20$  and  $\pm 30$  minutes scenarios lead to a decrease of 3.8%, 8.9% and 13.9% in the number of aircraft utilized, respectively. Along with this reduction, a decrease in the anticipated profit can occur since fewer aircraft are used to cover all flight legs and the number of short turns is variable. Notice that the number of iterations performed increases slowly over the four scenarios and that the heuristic linear relaxation solution is obtained quite rapidly (in less than 7 minutes).

Similar observations on the anticipated profit and the number of aircraft can be made about the integer solution part. Moreover, observe that the number of short time connections used is reasonable and that the number of branch-and-bound nodes is relatively small for each scenario. The latter can be explained by the fact that the solution to the linear relaxation of the Set Partitioning formulation provides an excellent bound for vehicle routing problems, which are similar to the DARSP, leading to an efficient branch-and-bound algorithm ([Bramel and Simchi-Levi 1993](#)).

Finally, the global information part shows the efficiency of the solution process. The test problems were solved in reasonable times. The solutions of the first two scenarios are optimal with respect to both the number of aircraft and the anticipated profit. For the last two scenarios, they are optimal with respect to the number of aircraft and show small integrality gaps on the anticipated profit (less than 0.5%).

**European Carrier Test Problems:** The second set of data presented in Table 2 corresponds to a typical weekday for a medium haul fleet of a European airline. The objective is to maximize the anticipated profit. The total CPU times to solve the test problems vary between 10 and 45 minutes on a HP 9000/720 workstation (66.1 Specfp92, 38.5 Specint92).

The tests were separated into three parts. In the first part, two scenarios with the same aircraft availability as the airline's solution were solved. The first of these scenarios considered fixed departure times while the second considered departure time windows of  $\pm 10$  minute width. In the second part, several scenarios were solved with departure time windows of  $\pm 10$  minute width and a total number of 51 available aircraft, but with different aircraft type availability. The goal of the last part was to optimize the fleet composition of the airline. A single scenario with departure time windows of  $\pm 10$  minute width was solved where the total number of available aircraft was fixed at 51 and the number of available aircraft of each type was unconstrained.

Figure 2 shows the profit improvement over the airline's solution of the different scenarios solved by GENCOL. The first column gives the aircraft types while the second indicates the number of aircraft of each type utilized in the airline's solution. The other columns present profit improvements obtained with GENCOL when the fleet size varied. For each scenario, the width of the departure time windows is given in the last row. The fleet size changes are provided in the corresponding aircraft type row (for example, a +2 in the B737 row of a scenario column means that 11+2 aircraft of type B737 are available for this scenario while a -2 in the A320 row means that 23 - 2 aircraft of type A320 are available). The last column corresponds to the fleet composition optimization scenario and the number of aircraft of each type utilized in the GENCOL solution is given directly on each row.

As illustrated in Figure 2, solving the problem with GENCOL using the same fleet as the one for the airline's solution and fixed departure times has led to a profit improvement of 6.8%. When allowing for flexible departure times (that is, departure time windows of  $\pm 10$  minute width), the improvement increases to 11.2%. The second part of the tests presents similar profit improvements (between 10.9% and 16.1%) when the composition of the fleet is modified slightly. Finally, the last scenario shows the influence of the fleet composition on the profit improvement. For departure time windows of  $\pm 10$  minute width, the profit improvement increases from 11.2% to 21.9% when GENCOL was allowed to determine the fleet composition.

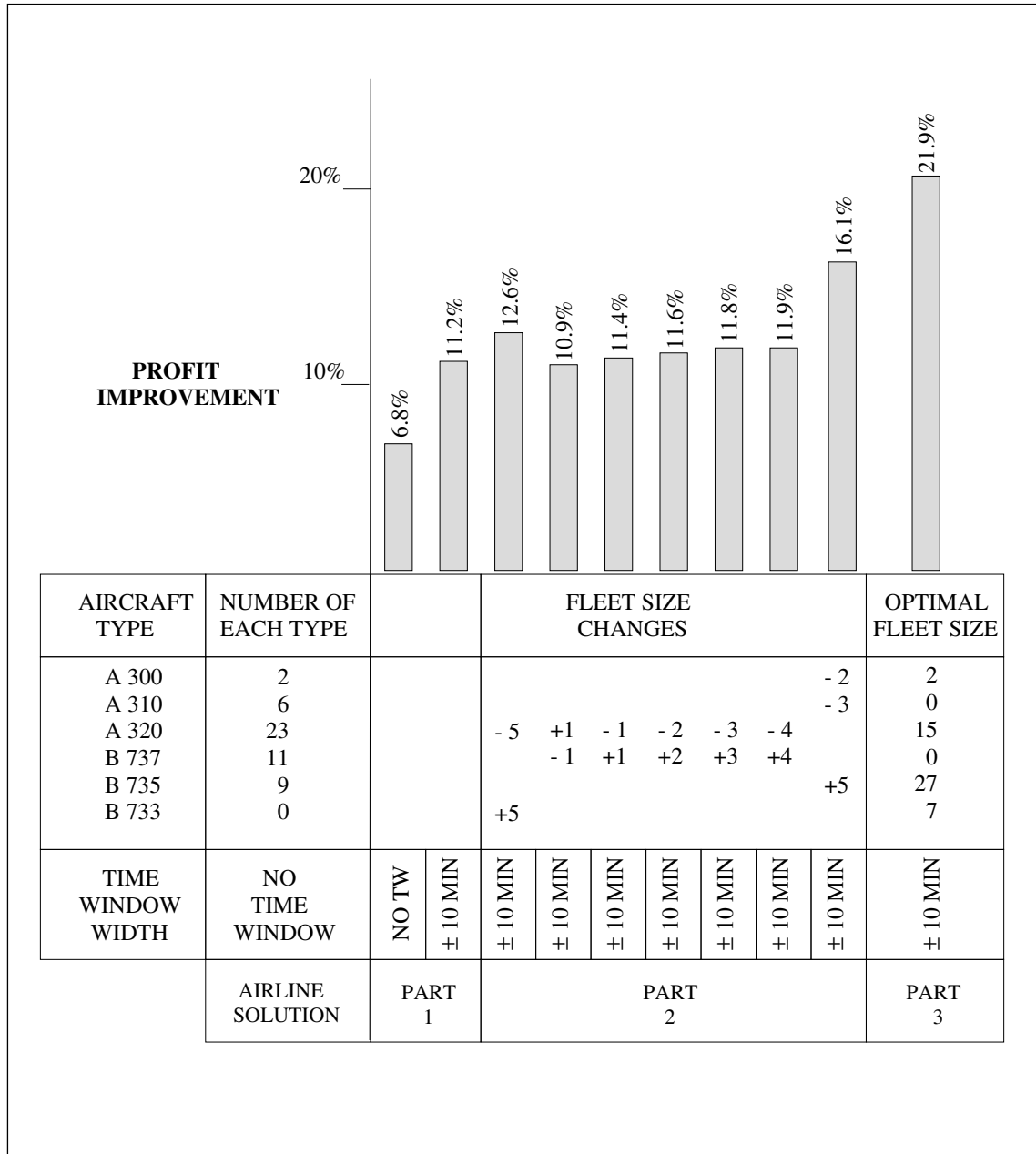


Figure 2: GENCOL's results on the European carrier data.

## 8 Conclusions

In this paper we have studied the daily aircraft routing and scheduling problem (DARSP) which consists of finding a fleet schedule that maximizes profits given a heterogeneous fleet of aircraft, a set of operational flight legs over a one-day horizon, departure time windows, durations and profits according to the aircraft type for each flight leg. We have presented two equivalent formulations for this problem: a multi-commodity network flow formulation previously described in the DARSP literature and a Set Partitioning type formulation recently utilized in the context of the vehicle routing problem with time windows. We have described the network structure of the subproblem when a column generation technique is applied to solve the linear relaxation of the Set Partitioning type formulation or a Dantzig-Wolfe decomposition approach is used to solve the linear relaxation of the time constrained multi-commodity network flow formulation.

We have used a branch-and-bound algorithm to obtain integer solutions. We have presented optimal branching strategies compatible with the column generation technique. These strategies were developed by exploiting the relationship between the two formulations. Finally, computational results obtained on real data provided by two airlines, one North American and the other European, have shown that branching on flow variables was very effective. They have also illustrated that the Dantzig-Wolfe decomposition or the column generation technique can be used to solve real-world problems in reasonable computational times, leading to substantial profit improvements.

**Acknowledgements:** This research was supported by the Quebec Government (Programme Synergie du Fonds de Développement Technologique) and by the Natural Sciences and Engineering Council of Canada. Marius M. Solomon was partially supported by the Patrick F. and Helen C. Walsh Research Professorship. We would also like to thank the management of Ad Opt Technologies, in particular Pierre Trudeau, and the programming team from GERAD consisting of Johanne Gilbert and Ianick Gentes. Finally, the authors wish to thank the Editor, the Associate Editor and two referees whose suggestions have improved the paper.

## 9 References

- ABARA, J. (1989), Applying Integer Linear Programming to the Fleet Assignment Problem, *Interfaces* 19, 20-28.
- AGIN N. and D. CULLEN (1975), An Algorithm for Transportation Routing and Vehicle Loading, *Logistics*, Edited by M. Geisler, North Holland, Amsterdam, 1-20.
- AHUJA, R.K., T.L. MAGNANTI and J.B. ORLIN (1993), *Network Flows*, Prentice Hall, New-Jersey.
- BARNHART, C., E. JOHNSON, R. ANBIL and L. HATAY (1991), A Column Generation Technique for the Long-Haul Crew Assignment Problem. *Optimization in Industry 2: Mathematical Programming and Modeling Techniques in Practice*, T.A. Ciriani and R. Leachman (eds.), John Wiley and Sons, 7-22.
- BRAMEL, J. and D. SIMCHI-LEVI (1993), *On the Effectiveness of Set Partitioning Formulations for the Vehicle Routing Problem*. Working Paper, Graduate School of Business, Columbia University.
- CHVÁTAL, V. (1983), *Linear Programming*, W.H. Freeman and Company, New-York.
- CPLEX Reference Manual (1993), *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, CPLEX Optimization, Inc., Incline Village, NV 89451-9436, U.S.A.
- DESROCHERS, M., J. DESROSIERS and M.M. SOLOMON (1992), A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows, *Operations Research* 40, 342-354.
- DESROCHERS, M. and F. SOUMIS (1988a), A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows. *INFOR* 26, 191-212.
- DESROCHERS, M. and F. SOUMIS (1988b), A Reoptimization Algorithm for the Shortest Path Problem with Time Windows. *European Journal of Operational Research* 35, 242-254.
- DESROSIERS, J., Y. DUMAS, M.M. SOLOMON and F. SOUMIS (1994), Time Constrained Routing and Scheduling, *Handbooks in Operations Research and Management Science*, Volume 8 on Network Routing, M. Ball, T. Magnanti, C. Monma and G. Nemhauser (eds.), Elsevier Science Publishers B.V., 35-139.

- DESROSIERS, J., F. SOUMIS and M. DESROCHERS (1984), Routing with Time Windows by Column Generation. *Networks* 14, 545-565.
- DESROSIERS, J., P. PELLETIER and F. SOUMIS (1983), Plus Court Chemin avec Contraintes d'Horaires. *RAIRO* 17, 357-377. (in French)
- ETSCHMAIER, M.M. and D.F.X. MATHAISEL (1984), Aircraft Scheduling: The State of the Art, *AGIFORS* 24, 181-225.
- GÉLINAS S., M. DESROCHERS, J. DESROSIERS and M.M. SOLOMON (1995), A New-Branching Strategy for Time Constrained Routing Problems with Application to Backhauling, *Annals of Operations Research* (forthcoming).
- HANE, C., C. BARNHART, E.L. JOHNSON, R. MARSTEN, G.L. NEMHAUSER and G. SIGISMONDI (1993), *The Fleet Assignment Problem: Solving a Large-Scale Integer Program*. Industrial and Systems Engineering Reports Series, COC-92-04, Georgia Institute of Technology.
- IOACHIM, I., S. GÉLINAS, J. DESROSIERS and F. SOUMIS (1994), *A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs*. Les Cahiers du GERAD G-94-24, École des Hautes Études Commerciales, Montréal, Canada.
- JARVIS, J.J. (1969), On the Equivalence between the Node-Arc and Arc-Chain Formulations for the Multi-Commodity Maximal Flow Problem, *Naval Research Logistics Quarterly* 16, 525-529.
- KOHL, N. and O.B.G. MADSEN (1993), *An Optimization Algorithm for the Vehicle Routing Problem with Time Windows based on Lagrangean Relaxation*. Research Report 17/1993, IMSOR, the Technical University of Denmark, Lyngby, DK-2800, Denmark.
- LASDON, L.S. (1970), *Optimization Theory for Large Systems*. Macmillan, New-York.
- LEVIN, A. (1971), Scheduling and Fleet Routing Models for Transportation Systems, *Transportation Science* 5, 232-255.
- TOMLIN, J.A. (1966), Minimum-Cost Multicommodity Network Flows, *Operations Research* 14, 45-51.