View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Directory of Open Access Journals

Yugoslav Journal of Operations Research
15 (2005), Number 1, 25-48

THE OPERATIONAL FLIGHT AND MULTI-CREW SCHEDULING PROBLEM

Mirela STOJKOVIĆ, François SOUMIS

École Polytechnique de Montréal and GERAD
Montréal (Québec), Canada H3T 2A7
Mirela.Stojkovic@gerad.ca
Francois.Soumis@gerad.ca

Presented at XXX Yugoslav Simposium on Operations Research
Received: December 2003 / Accepted: March 2004

Abstract: This paper introduces a new kind of operational multi-crew scheduling problem which consists in simultaneously modifying, as necessary, the existing flight departure times and planned individual work days (duties) for the set of crew members, while respecting predefined aircraft itineraries. The splitting of a planned crew is allowed during a day of operations, where it is more important to cover a flight than to keep planned crew members together. The objective is to cover a maximum number of flights from a day of operations while minimizing changes in both the flight schedule and the next-day planned duties for the considered crew members. A new type of the same flight departure time constraints is introduced. They ensure that a flight which belongs to several personalized duties, where the number of duties is equal to the number of crew members assigned to the flight, will have the same departure time in each of these duties. Two variants of the problem are considered. The first variant allows covering of flights by less than the planned number of crew members, while the second one requires covering of flights by a complete crew. The problem is mathematically formulated as an integer nonlinear multi-commodity network flow model with time windows and supplementary constraints. The optimal solution approach is based on Dantzig-Wolfe decomposition/column generation embedded into a branch-and-bound scheme. The resulting computational times on commercial-size problems are very good. Our new simultaneous approach produces solutions whose quality is far better than that of the traditional sequential approach where the flight schedule has been changed first and then input as a fixed data to the crew scheduling problem.

Keywords: Crew recovery, flight scheduling, aircraft routing, shortest path, time windows, column generation.

1. INTRODUCTION

The optimization approach recently proposed by Stojković and Soumis (2001) treats simultaneously the single-crew and the flight scheduling problems. The approach allows the modification of one-day planned individual activities (duties) for a set of selected pilots in a given category, i.e. captains, by permitting the delay of certain flights, when necessary, while still preserving the predefined aircraft itineraries. Important passenger connections are preserved by adding precedence constraints on departure times of corresponding flights. The aircraft maintenance schedule can be respected by imposing in advance a maximum acceptable delay on some flights. The objective is to minimize the number of uncovered flights and the total delay of rescheduled flights in the considered day of operations as well as the total number of crew members whose next-day duties must be changed due to the proposed modifications. The model solves the multi-crew rescheduling problem restricted to the special case where duty modifications apply to the whole crew together.

This paper treats the general form of the multi-crew rescheduling problem, where duty modifications are not necessarily identical for individual members of a crew. We consider the flight attendant problem where positions are interchangeable. The multi-crew problem without interchangeable positions may be decomposed into one problem per position and solved by the approach presented in Stojković and Soumis (2001).

The problem considered consists in covering each of the given flights from the considered day of operations with a predetermined number of crew members while permitting the delay of some flights when necessary. Each flight belongs to several personalized duties, where the number of duties corresponds to the number of crew members required to cover the flight. If a flight must be delayed, its new departure time must be the same for all crew members to whom this flight is assigned. These same flight departure time constraints represent a new problem feature with respect to the previous model developed by Stojković and Soumis (2001). Fixed aircraft itineraries, some important passenger connections and the aircraft maintenance schedule can be preserved by imposing corresponding precedence constraints, such as previously described by Stojković and Soumis (2001). If needed, some supplementary precedence constraints which ensure the feasibility of itineraries of the technical personnel may be imposed in the master problem. In addition, the width of the initial time windows associated with the considered flights may be reduced to respect the maximum duty duration of the technical personnel. The multi-crew scheduling introduces a new difficulty regarding flight covering. Two models can be considered. The first model allows partial covering of flights, while the second model requires covering of flights by a complete crew. Consequently, two forms of the cost for uncovered flights are considered: a linear cost on the number of missing crew members in the first model and a cost for each flight not completely covered in the second model. The other terms in the objective are to minimize the total delay of considered flights and the number of crew members whose next-day duties must be changed due to the proposed modifications.

The contributions of this work are as follows: First, we allow splitting of a crew during a day of operations, where it is more important to cover the flights than to keep crew members together through the day. Keeping them together may not even be possible given the different activities that they have already performed and the different future activities still to be performed. Even if they were planned to work together, some crew teams have already been broken due to past disturbances, replacements due to sicknesses, training periods, medical exams or any other unforeseen events. We modeled the same flight departure time constraints and proposed a multi-commodity flow formulation that includes this new type of constraint. Second, we developed two models that treat the cost of uncovered flights in different ways. Third, to efficiently solve the problem when using the second model, we upgraded the branching method used in the first model by introducing a new type of branching decision. Finally, we implemented and tested both models. The computational experiments confirm the efficiency of branching strategies used to solve the real-size problems for both models.

The remainder of the paper is organized into four sections. Section 1 presents a mathematical model, the solution process to solve it, and describes two modeling approaches that we propose. Section 2 shows the computational results obtained on several test problems and describes the new branch-and-bound strategy developed for the second model. Finally, conclusions and perspectives are discussed in Section 3.

2. PROPOSED MODEL

The problem is to cover with available crew members a set of flights and to determine their new departure times. Crew members planned to be together on a flight may be reassigned to different flights. A time window associated with a flight is defined according to commercial and operational constraints and possibly reduced to respect the maximum duty duration of the technical personnel. If the time window is reduced to a point, then the flight departure time is fixed. Otherwise, the flight departure time is flexible. The set of available crew members comprises crew members whose duty had been partially performed before the disturbance took place (active crew members or reserves on duty). Crew members on rest and reserves on call are excluded from the set because each of them must be phoned first before the proposed assignment may be considered as accepted (they may not answer the phone or may refuse the assignment). Thus, the problems that can be resolved by using the rest crew members and reserves on call remain currently in the domain of a crew operator's responsibility. Once the resolved problems are removed, the residual uncovered flights have to be covered. This problem represents a generalization of the problem solved by Stojković and Soumis (2001), where a single crew member was required to cover each flight. In the same paper the authors also present a review of the literature on the operational crew scheduling problem. The corresponding publications are: Stojković, Soumis and Desrosiers (1998), Lettovský (1997), Lettovský, Johnson and Nemhauser (1999), Luo and Yu (1998a; 1998b), Monroe and Chu (1995) and Wei, Yu and Song (1997).

2.1. Notation

Let F, indexed by f, represent the set of flights to be covered during the day of operations. Each considered flight must be covered by a predefined number of crew members. Let n_f denote the number of crew members required to cover a flight $f \in F$. Flight $f \in F$ can be represented by n_f copies, referred to as tasks, which must be covered by a single crew member. Thus, with each flight $f \in F$ are associated one ($n_f = 1$) or more ($n_f > 1$) tasks identical to the original flight, except that they require only a single crew member to be covered. Remark 1 in Section 1.2 discusses the model without replicating flights into tasks. It is shown that it is not easy to obtain a linear programming formulation of the problem stated in this way.

Let *N*, indexed by *i*, represent the set of all tasks to be covered by a single crew member, $|N| = \sum_{f \in F} n_f$. Obviously, we have that $|F| \leq |N|$. Let $N_f \subset N$ represent the set of all tasks *i* being derived from flight $f \in F$. We have that $|N_f| = n_f$.

Denote by $[a_f, b_f]$ a time window associated with flight $f \in F$. Denote by T_f, $f \in F$, a time variable representing the amount of delay of flight $f \in F$, $0 \leq T_f \leq (b_f - a_f)$. Each considered flight $f \in F$ has either a fixed $(b_f = a_f)$ or a flexible $(b_f > a_f)$ departure time. Denote by $F_{flex} \subset F$ the set of all flexible scheduled flights *f* and by $F'_{flex} \subset F_{flex}$ the set of flexible scheduled flights *f* that must be covered by more than a single crew member $(n_f > 1)$. Let *B* be a set of pairs of flights (f, h), $f, h \in F$, where flight $h \in F$ follows flight $f \in F$ in an aircraft itinerary or when there is an important group of passengers connecting from *f* to *h*. Some flight precedence constraints must be imposed to ensure the feasibility of aircraft itineraries and passenger connections. Let d_{fh} be the minimum time required between *f* and *h*. As recently described by Stojković and Soumis (2001), only if both *f* and *h* have flexible departure times must the flight precedence constraint be explicitly imposed. Let *E* be the set of pairs of flexible scheduled flights with a precedence relation between them, $E = B \cap (F_{flex} \times F_{flex})$. The case when crew members are grouped into inseparable sub-teams can be treated also. In this case one task corresponds to a requirement for a sub-team for a flight. Even if this more general case can be treated by the same model, we use the language for a singleperson sub-team. We assume here that the sub-teams are of the same size and interchangeable. When it is not the case, it must be remembered from the introductory section that the problem can be decomposed into one problem per sub-team and solved by the approach presented in Stojković and Soumis (2001).

Let *K*, indexed by *k*, be the set of crew members on duty. With each crew member *k* is associated a graph $G^k = (V^k, A^k)$, where V^k is the set of nodes and A^k is the set of arcs. The set of nodes V^k contains the source node *o*(*k*), the sink node *d*(*k*) and the set of nodes N^k, $N^k \subset N$, that can be visited by a path of commodity *k*. Each node in N^k corresponds thus to a task that can be assigned to crew member $k \in K$. Hence, $V^K = N^k \cup \{o(k), d(k)\}$ and $N = \bigcup_{k \in K} N^k$. The set of arcs A^k consists of beginning, ending and connection arcs and an origin-destination arc, (*o*(*k*), *d*(*k*)). A path in G^k originates at *o*(*k*) and ends at *d*(*k*). The path containing only the origin-destination arc corresponds to a crew member with no further activities assigned during the day of operations. All the other feasible paths in G^k correspond to feasible duties for crew member *k*. The constraints concerning the maximum duration of a modified duty and the minimum rest between a modified duty and the next planned duty of crew member *k* are respected by calculating, for each of them, the latest moment when a duty for crew member *k* may terminate. The smallest of the two values is then associated with the the sink node $d(k)$. We consider that a duty for crew member k is feasible if the corresponding path respects the minimum briefing, debriefing and crew connection times and terminates no later than the time associated with the sink node $d(k)$. These are verified during the construction of the set of arcs A^k, when an arc is included in the set only if the corresponding value is respected. More details about the sets of nodes and arcs are given in Stojković and Soumis (2001).

Let $X_{ij}^k, k \in K, (i, j) \in A^k$ represent a binary network flow variable which takes value 1 if arc (i, j) was used in the solution duty for crew member k and 0 otherwise. Finally, denote by $T_i^k, k \in K, i \in N^k$ a time variable which represents a departure time of task i if it is performed by crew member k and 0 otherwise.

A cost c_{ij}^k is associated with each arc $(i, j) \in A^k$. To encourage a new duty of crew member $k \in K$ to terminate at the planned destination airport, a penalty cost $c_{i,d(k)}^k$, $(i, d(k)) \in A^k, i \in N^k \cup o(k)$, is associated with the corresponding arcs if the destination airport of task i does not correspond to the planned destination airport for crew member k. There is also a fixed unit cost u_i associated with each minute of delay of task $i \in N$. If u_f represents a unit cost associated with a minute of delay of flight $f \in F$, then the unit cost associated with its task i is given by $u_i = u_f / n_f$. It is modeled as a node cost.

2.2. Column Generation Formulation

The problem is mathematically formulated as the integer nonlinear multi-commodity network flow model with time windows. The mathematical formulation is identical to that presented in Stojković and Soumis (2001), except for a new set of same flight departure time constraints. These constraints are written as:

$$\sum_{k \in K} T_i^k = a_f + T_f, \qquad \forall f \in F'_{flex}, \forall i \in N_f. \tag{1}$$

Obviously, all n_f tasks derived from a flight $f \in F'_{flex}$ must have the same departure time.

To obtain an optimal integer solution of the problem, a column generation approach embedded within a branch-and-bound procedure is used. First, the linear relaxation of the problem is solved by Dantzig-Wolfe decomposition/column generation approach. Second, to obtain an optimal integer solution the previous step is incorporated into a branch-and-bound scheme, where each such linear relaxation solution gives a lower bound for the explored branch. A specialized branch-and bound technique, which uses particular characteristics of the problem, is used to obtain an optimal integer solution.

The decomposition scheme comprises a master problem and a subproblem for each crew member k.

Master Problem: Let Ω^k, indexed by p, be the set of all extreme points of subproblem $k \in K$. Each extreme point corresponds to a feasible path (feasible duty) in G^k. Let θ_p^k represent the master problem variable associated with the selection of path $p \in \Omega^k$ for crew member k, with cost c_p^k. Denote by

$$(x_p^k, t_p^k) = (x_{ij,p}^k, t_{i,p}^k), \quad (i,j) \in A^k, i \in N^k, p \in \Omega^k \tag{2}$$

the coordinates of extreme point p of subproblem k.

In the master problem, let $a_{i,p}^k, i \in N, p \in \Omega^k, k \in K,$ be a coefficient corresponding to the flight covering constraints and $b_{i,p}^k, i \in N, p \in \Omega^k, k \in K$, be a coefficient corresponding to the same flight departure time constraints and the flight precedence constraints.

Using this notation, the master problem can be written as:

$$\text{minimize} \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \theta_p^k + \sum_{i \in N} u_i \sum_{k \in K} \sum_{p \in \Omega^k} b_{i,p}^k \theta_p^k \tag{3}$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{i,p}^k \theta_p^k = 1, \quad \forall i \in N \tag{4}$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} b_{i,p}^k \theta_p^k = a_f + T_f, \quad \forall f \in F'_{flex}, \forall i \in N_f \tag{5}$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} (b_{j,p}^k - b_{i,p}^k) \theta_p^k \geq d_{fh}, \quad \forall (f,h) \in E, \forall (i,j) \in N_f X N_h \tag{6}$$

$$\sum_{p \in \Omega^k} \theta_p^k = 1, \quad \forall k \in K \tag{7}$$

$$\theta_p^k \geq 0, \quad \forall k \in K, \forall p \in \Omega^k \tag{8}$$

$$X_{ij}^k = \sum_{p \in \Omega^k} x_{ij,p}^k \theta_p^k, \quad \forall k \in K, \forall (i,j) \in A^k \tag{9}$$

$$X_{ij}^k \text{ binary}, \quad \forall k \in K, \forall (i,j) \in A^k \tag{10}$$

The cost function (3) minimizes, respectively, the number of crew members whose nextday planned operations must be changed as a consequence of the proposed modifications to the given day of operations, and the total delay of considered flights. Constraints (4), (5) and (6) represent, respectively, the covering constraints, the same flight departure time constraints and the flight precedence constraints. Provided the path variable non-negativity constraints (8) are satisfied, the convexity constraints (7) indicate that exactly one path must be assigned to each crew member. Constraints (10) impose binary values for the flow variables, expressed in terms of the path variables and extreme points by constraints (9). The model defined by (3)-(10) does not allow for the uncovering of flights. The problem of uncovered flights is discussed in Section 1.3.

The only integer variables in the formulation (3)-(10) are X_{ij}^k variables. Thus, the linear relaxation of the master problem is obtained by eliminating constraints (9) and (10).

The integer requirement is imposed on the flow variables originating from the integer nonlinear multi-commodity network flow formulation before applying Dantzig-

Wolfe decomposition. It will be shown later in the paper (Remarks 2 and 3) that this integer requirement can be replaced.

The presented model replicating each flight $f \in F$ that must be covered n_f times into n_f identical tasks that must be covered exactly once may seem quite artificial. The following remark addresses this question.

Remark 1. Even though covering n_f times a flight $f \in F$ seems more natural than deriving n_f tasks from each flight *f* and then covering them exactly once, the resulting model without replicating flights into tasks is far more complex.

In the model without replicating flights the covering constraints of the multi-commodity network flow formulation before applying Dantzig-Wolfe decomposition would be:

$$\sum_{k \in K} \sum_{h:(f,h) \in A^k} X_{f,h}^k = n_f, \quad \forall f \in F \tag{11}$$

It is easy to remark that the number of constraints (11) is reduced compared to the case with tasks derived from flights (|*F*| vs. |*N*| constraints). However, even without replicating flights into tasks we still need the same departure time constraints. If D_f^k is the variable representing the departure time of flight $f \in F$ if it is performed by crew member $k \in K$ and 0 otherwise, constraints (1) from the formulation before applying Dantzig-Wolfe decomposition must be replaced by the more complex constraints:

$$D_f^k = (a_f + T_f) \sum_{h:(f,h) \in A^k} X_{f,h}^k, \quad \forall f \in F'_{flex}, \forall k \in K\,. \tag{12}$$

The same departure time constraints (12) are still needed, since the departure time of flight $f \in F$ must be the same for each crew member $k \in K$ covering the flight. The number of constraints (12) is $(|F'_{flex}| * |K|)$. This is significantly higher than the number of constraints (1), which is equal to $\sum_{f \in F'_{flex}} n_f$. Furthermore, since the left side of equation (12) is equal to the flight departure time only if crew member $k \in K$ covers flight $f \in F$ and zero otherwise, the equality does not hold without multiplying its right side by $\sum_{h:(f,h) \in A^k} X_{fh}^k$. The resulting constraints (12) are thus nonlinear. Obviously, we prefer solving the linear formulation with n_f tasks representing a flight $f \in F$.

Subproblems: The objective of subproblem *k* is to produce the minimum reduced cost column generated by network G^k. A reduced cost of a path in the subproblem's network is obtained by using dual variables associated with the master problem constraints. Let $\boldsymbol{\alpha} = \{\alpha_i \mid i \in N\}$, $\boldsymbol{\beta} = \{\beta_i \mid i \in N\}$, $\boldsymbol{\gamma} = \{\gamma_{(i,j)} \mid (i,j) \in N_f X N_h, (f,h) \in E\}$ and $\boldsymbol{\delta} = \{\delta^k \mid k \in K\}$ be the vectors of dual variables associated with constraint sets (4), (5), (6) and (7), respectively. We must note that dual variable β_i is associated with each task $i \in \bigcup_{f \in F'_{flex}} N_f$, since the same flight departure time constraints are presented only for the tasks derived from the flexible scheduled flights that must be covered by at least two crew members. However, by assuming that dual variable β_i is zero for a task $i \in \bigcup_{f \in F \setminus F'_{flex}} N_f$, we may associate dual variable β_i with each task $i \in N$. Thus, the vector $\boldsymbol{\beta}$ of dual variables β_i is given for all $i \in N$. As in the original problem formulation, two types of variables are presented in subproblem *k*. These are flow variables $X_{ij}^k, (i,j) \in A^k$ which are equal to 1 for arcs in the shortest path and 0 otherwise, and time variables T_i^k which represent the departure time of task $i \in N^k$. To facilitate the notation, denote by *W* the set of all pairs of tasks on which the precedence constraints (6) are imposed. We have that $W = \bigcup_{(f,h) \in E} (N_f \times N_h)$. Finally, denote by $W^k \subset W$ a set of pairs of tasks belonging to subproblem $k \in K$ on which the flight precedence constraint must be imposed. At least one of the tasks in each pair must belong to the set of nodes N^k. Thus, $W^k = \{(i,j) \in W \mid i \in N^k \vee j \in N^k\}$ and $W = \bigcup_{k \in K} W^k$. The subproblem *k* is identical to that presented in Stojković and Soumis (2001) except for the cost function that is written as:

$$\text{minimize} \sum_{(i,j) \in A^k} (c_{ij}^k - \alpha_i) X_{ij}^k + \sum_{i \in N^k} (u_i - \beta_i) T_i^k - \sum_{(i,j) \in W^k} \gamma_{(i,j)} (T_j^k - T_i^k) - \delta^k . \quad (13)$$

The subproblem *k*, which represents a minimum cost path problem with time windows and linear costs on flow and time variables, is solved by the same optimal dynamic programming algorithm for acyclic networks that had been used in Stojković and Soumis (2001).

Integer Solutions: Once the linear relaxation of the master problem is solved, the process is embedded within a branch-and-bound scheme to obtain an optimal integer solution of the problem. To perform this task, we define a binary branch-and-bound tree whose root corresponds to the linear relaxation of the master problem, defined by (3)-(8). The other nodes are created by adding branching decisions to both the master problem and the subproblem. Constraints (9) and (10), eliminated while searching for the linear relaxation of the problem, must be imposed in order to obtain integer flow variables X_{ij}^k.

It may seem more natural to branch on path variables θ_p^k from the master problem. The following remarks help to understand the elimination of this choice.

Remark 2. The integrality requirement on X_{ij}^k can be replaced by the integrality requirement on θ_p^k.

While an optimal solution with $\theta_p^k, k \in K, p \in \Omega^k$ binary implies binary flow variables $X_{ij}^k, k \in K, (i,j) \in A^k$, an optimal solution with binary $X_{ij}^k, k \in K, (i,j) \in A^k$ does not imply binary path variables $\theta_p^k, k \in K, p \in \Omega^k$. However, the formulation (3)-(10) possesses an optimal solution with $\theta_p^k, k \in K, p \in \Omega^k$ binary (Proposition 1, Stojković and Soumis (2001)).

Yet we do not branch on path variables. One of the reasons is because there are too many of them. The other one is that a branching decision $\theta_p^k = 0$ is very weak. In addition, imposing such a decision makes the associated branching tree unbalanced. The next remark introduces a choice of better branching variables than θ_p^k and X_{ij}^k variables. Let $X_{ij} = \sum_{k\in K} X_{ij}^k$, $(i, j) \in \bigcup_k A^k$.

Remark 3. The integrality requirement on X_{ij}^k variables from the original formulation of the problem can be replaced by the integrality requirement on X_{ij} variables.

In practice, we branch on flow variables X_{ij} rather than on the variables X_{ij}^k. Namely, even if tasks *i* and *j* can be found in several subproblems $k \in K$, each of these tasks represents the master problem's task that has to be covered exactly once. Obviously, if all X_{ij}^k are binary, then X_{ij} are also binary. The opposite statement is also true: if all X_{ij} variables are binary in an optimal solution, then the variables X_{ij}^k are binary too. In fact, the set of nodes $\bigcup_{k\in K} V^k$ is covered by a set of disjoint paths when X_{ij} variables are binary. Each of these paths originates from a single origin node *o*(*k*). The variables X_{ij}^k for the arcs involved in this path are equal to 1 and the others are equal to 0.

The number of X_{ij} variables, $(i, j) \in \bigcup_k A^k$, is smaller than the number of X_{ij}^k variables and significantly smaller than the number of θ_p^k variables. In addition, the corresponding branching tree is balanced. Consequently, the constraint sets (9) and (10) will be replaced by the following constraints:

$$X_{ij} = \sum_k \sum_{p\in\Omega^k} x_{ij,p}^k \theta_p^k, \quad \forall (i, j) \in A^k \tag{14}$$

$$X_{ij} \text{ binary}, \quad \forall (i, j) \in \bigcup_k A^k \tag{15}$$

For the present application, we used a branching technique involving decisions both on the time variables T_f, $f \in F$ and the flow variables $X_{ij}, (i, j) \in \bigcup_k A^k$. The technique is a modification of the technique presented in details in Stojković and Soumis (2001). The method consists in imposing decisions first on the time variables. When it becomes impossible to impose any more decisions on the time variables, but some flow variables in the master problem are still fractional, then the search for the optimal integer solution continues by imposing decisions on the flow variables.

2.3. Cost Modeling

The mathematical model presented in Section 1.2 needs to be completed, since in many cases it is impossible to cover the flights with the available crew members. Two possibilities of dealing with the problem of uncovered flights are considered. The first option permits the partial coverage of flights, i.e. the production of a solution in which the flights are not necessarily covered by a complete crew. Such a solution may be accepted by an airline company if the number of covered crew positions (tasks) per flight is sufficient to operate it. Even if few flights have less than the minimum number of crew members required, crew operators may manage to find the number of crew members needed to meet the required minimum by using resources outside of the set of crew members that we considered. We propose Model 1 in case partially covered flights are acceptable.

The second option does not permit partial coverage; either a flight is covered by a complete crew or all its tasks are uncovered. We propose Model 2 in case the complete crew is required to operate a flight.

To implement Model 1, we introduced an artificial crew member in the set of available crew members defined in Section 1. We added the artificial arcs to create paths which begin at the artificial source node and then, after visiting a single task $i \in N$, return to the artificial sink node. A very large cost, corresponding to the penalty for the uncovering of each task i, is associated with each of these paths.

To implement Model 2, we introduced an artificial crew in the set of available crew members. We added the artificial arcs to create paths which begin at the artificial source node and then, after visiting all n_f tasks derived from flight $f \in F$, return to the artificial sink node. A very large cost, corresponding to the penalty for the uncovering of all n_f tasks, is associated with each of these paths. The artificial commodity is excluded from the constraint set (7) in both Model 1 and Model 2.

3. NUMERICAL EXAMPLE

Models 1 and 2 have been implemented and tested on four input data sets, named respectively Problem 1, Problem 2, Problem 3 and Problem 4. All of the considered flights are domestic US flights. The values of the briefing and debriefing time, the maximum duty duration and the minimum crew connection time were taken from a collective agreement. In the absence of information regarding planned passenger itineraries only the set of flight precedence constraints imposed to ensure the feasibility of aircraft itineraries has been considered in our numerical experiments. Deadheads are not allowed, either on the company's flights or on flights from other companies. It follows that flight over-covering is not allowed. A hypothetical situation, where the hub airport is closed in the afternoon peak hour, is considered as the source of disturbances in all four cases. As a consequence, all flights planned to land at this airport before the moment of its reopening were directly influenced by the given disturbance. One of these directly influenced flights was canceled in Problems 2, 3 and 4. New departure times were fixed for the remaining delayed flights. If, due to introduced delays, a ground time between a delayed flight and its succeeding flight from the same aircraft itinerary became smaller than the minimum required value, the succeeding flight was consequently delayed in order to meet the required minimum. New departure times for delayed succeeding flights were fixed, too. All these initially delayed flights are considered as the fixed scheduled flights. The flights planned to land at an airport different from the hub, and which had already departed at the moment the disturbance was announced, are also considered as the fixed scheduled flights. Originally fixed departure times for the rest of the planned flights have been transformed to flexible ones. The maximum width of a time window, which corresponds to the maximum allowed delay, is fixed to 1 hour. Crew members whose planned assignments include delayed flights, as well as crew members assigned to involved aircraft, are considered as candidates for modifications. Table 1 highlights the characteristics of the considered problems. For the given set of problems the percentage of flexible scheduled flights varies from 61% (Problem 4) to 71% (Problem 1).

Table 1: Schedule Characteristics

Problems	Problem 1	Problem 2	Problem 3	Problem 4
Aircraft	13	28	58	79
Flights:				
Total Number	24	66	131	190
Fixed Departure Time	7	20	50	76
Fixed Departure Time	17	46	81	114

Several test problems were further generated for each of the four problems. Tests generated from the same problem differed by crew size and thus by the total number of tasks. The characteristics of each of these test problems are presented in Table 2. The test identifier is composed of two digits. The first digit left of the dot is the problem identifier, while the second digit represents the crew size (number of crew members per crew). For example, from Problem 1 we created 6 tests representing crews of 2 to 7 members. The corresponding identifiers are Test 1.2 to Test 1.7. Derived tests from Problems 2 to 4 are identified in the same way.

Table 2: Test characteristics

Tests	Crew Size	Tasks	Crew Members
Problem 1			
Test 1.2	2	48	14
Test 1.3	3	72	21
Test 1.4	4	96	28
Test 1.5	5	120	35
Test 1.6	6	144	42
Test 1.7	7	168	49
Problem 2			
Test 2.2	2	132	43
Test 2.3	3	198	61
Test 2.4	4	264	79
Test 2.5	5	330	97
Problem 3			
Test 3.2	2	262	84
Test 3.3	3	393	126
Test 3.4	4	524	168
Problem 4			
Test 4.2	2	380	118
Test 4.3	3	570	177

The increase in the crew size induces the increase in both the total number of involved crew members and the total number of tasks needing to be covered. The first six tests created from Problem 1 correspond to small volume disturbances, while the other tests correspond to medium and high volume disturbances for a large fleet. Small volume disturbances happen frequently during the day of operations, while the other disturbances are far less frequent, the high volume disturbances involving a large number of affected crew members being very uncommon.

Table 3 reveals the results of the systematic repair procedure. It consists in first changing the flight schedule and then repairing the crew schedule by further delaying some of the input flexible scheduled flights without modifying the crew itineraries and the input aircraft itineraries. The first flight within a duty requiring a delay longer than permitted becomes uncovered, and so do the successive flights within the same duty. Thus the duty becomes infeasible. A duty may also become infeasible if the new flight schedule extends the duty duration beyond the imposed maximum. For each of the test problems shown in the first column of Table 3, the second column gives the total number of uncovered tasks and the third column gives the number of infeasible planned duties. If the last covered flight in an infeasible duty does not terminate at the planned final airport of the original duty, the corresponding crew member cannot continue her/his planned next-day activities. Such a crew member is referred to as misplaced crew member. The last column of the table shows the number of misplaced crew members. Results presented in Table 3 will be used as a reference point when assessing the quality of the proposed optimization approach.

Table 3: Solution preserving aircraft and crew itineraries

Tests	Uncovered Tasks	Infeasible Duties	Misplaced Crew Members
Problem 1			
Test 1.2	22	8	5
Test 1.3	34	12	7
Test 1.4	44	16	10
Test 1.5	56	20	12
Test 1.6	66	24	15
Test 1.7	78	28	17
Problem 2			
Test 2.2	26	9	6
Test 2.3	39	13	8
Test 2.4	52	17	10
Test 2.5	65	21	12
Problem 3			
Test 3.2	40	18	10
Test 3.3	60	27	15
Test 3.4	80	36	20
Problem 4			
Test 4.2	56	26	18
Test 4.3	84	39	27

3.1. Model 1

Model 1 has been implemented to improve the results presented in Table 3. Corresponding results are presented in Table 4.

Table 4: Model 1: Solution with crew schedule reoptimization

Tests	Uncovered Tasks	Misplaced Crew Members	Delayed Flights	Average Delay (min)
Problem 1				
Test 1.2	10	7	4	46
Test 1.3	16	10	4	46
Test 1.4	20	14	4	46
Test 1.5	26	17	4	46
Test 1.6	30	21	4	46
Test 1.7	36	24	4	46
Problem 2				
Test 2.2	7	7	8	38
Test 2.3	10	10	10	33
Test 2.4	13	13	11	30
Test 2.5	16	16	8	38
Problem 3				
Test 3.2	8	10	15	32
Test 3.3	12	15	14	34
Test 3.4	16	20	14	34
Problem 4				
Test 4.2	14	14	20	34
Test 4.3	21	21	20	34

The columns of Table 4 reveal the residual problems in terms of the number of uncovered tasks, the number of misplaced crew members, the number of flights delayed by the optimizer and the average delay in minutes. The table shows that the results obtained by applying our optimization model are far better than those presented in Table 3. The number of uncovered tasks is significantly reduced in all considered test problems, while the number of misplaced crew members is reduced in several test problems and either equal to or greater than the previous values for the remaining test problems. However, the average reduction in the number of uncovered tasks for all considered test problems is about 68%, which is significantly higher than the average increase in the number of misplaced crew members, which is only about 14%. Thus, the great majority of problems concerning the considered day of operations (uncovered flights) have been successfully solved, leaving some problems for the next day of operations (misplaced crew members). Note that the number of misplaced crew members could be significantly reduced by using deadhead flights to transport crew members between airports. However, selection of deadhead flights during irregular operations differs from that used in the planning phase, where a set of deadhead candidates may be preselected in advance. Irregular operations influence flight schedules for many airline companies (all companies visiting the involved airport, for example). Choosing potential deadhead flights before the flight schedule is settled for each company increases the dimensions of the problem to be solved, while inducing high risks of producing a solution which is infeasible in reality due to unavailability of deadhead flights used in the solution (delayed or cancelled flight, no set available, etc.). It is far less complex and thus more efficient and reliable to solve the problem without using any deadhead and to select afterwards only those needed either to cover the flights that remained uncovered after the reoptimization or to position misplaced crew members at the desired airports. The number of uncovered flights and misplaced crew members may be also reduced by using a priori selected reserve crew members. Even if it is easy to introduce reserves into the model, we do not use them in our experiments because we did not have any data on available reserve crew members. The related data is complex, consisting not only of the number of reserves available, but also of the status of each of them that depends on the already performed activities. Including anonymous reserves in the model does not reflect the reality, since the status of these reserves is not known. A duty tailored to an anonymous reserve crew member may not go with already performed activities of any real reserve crew member. On the other hand, it is very difficult to justify a choice of artificial reserve data including not only their number but also their status. Our choice is thus to use no reserve crew member while solving the problem.

The number of nodes in the corresponding graphs ranges from 78 (Test 1.2) to 926 (Test 4.3). The number of arcs ranges from 914 (Test 1.2) to 120036 (Test 4.3). The number of commodities, which equals the total number of considered real crew members plus one (because of the artificial crew member used to cover uncovered tasks), ranges from 15 (Test 1.2) to 178 (Test 4.3). Given a single day of operations to be solved, many time requirements for a duty, a pairing, or a monthly block are regrouped into a single constraint and modeled by using a single resource. The other constraints are embedded into the graph definition. The set of master problem constraints consists of the flight precedence, the same flight departure time and the flight covering constraints. The number of flight precedence constraints ranges from 9 (the tests derived from Problem 1) to 50 (the tests derived from Problem 4). The number of same flight departure time constraints in a particular test problem is equal to the number of flights with flexible departure times in that problem (Table 1) multiplied by the crew size required for that problem (Column 2, Table 2). It ranges from 34 (Test 1.2) to 342 (Test 4.3). The number of covering constraints in a particular test problem, which ranges from 62 (Test 1.2) to 747 (Test 4.3), is the sum of the number of tasks and the number of involved crew members (Columns 3 and 4, Table 2), because exactly one path must be assigned to a real crew member (constraints (7)). If no duty is assigned to a real crew member, the path composed of only the origin-destination arc will be assigned to her/him. The total number of master problem constraints increases significantly as the crew size increases (from 105 for Test 1.2 to 1139 for Test 4.3) and the corresponding problems become more and more difficult to solve.

The test problems were solved using a specialized version of the GENCOL optimizer. The linear programs were solved using CPLEX Linear Optimizer 6.5. The primal simplex method was used to solve the linear relaxation of the master problem. The best-first search method was applied when searching for the optimal integer solution. All problems were solved to optimality. These computational experiments were performed by using a single processor of Enterprise 10000 workstation (64 CPU, 400 MHz, 64 G Ram).

Table 5 gives computational results. The second column of the table reveals the total number of branch-and-bound nodes explored while searching for the optimal solution. The third column gives the integrality gap between the optimal integer solution and the linear relaxation solution in percentage points. The last column contains the total computational time in seconds.

Table 5: Model 1: Computing times - optimal branching strategy

Tests	BB Nodes	Integrality Gap (%)	CPU (sec)
Problem 1			
Test 1.2	12	0.045	1
Test 1.3	24	0.039	4
Test 1.4	64	0.023	14
Test 1.5	197	0.021	54
Test 1.6	422	0.015	196
Test 1.7	340	0.015	252
Problem 2			
Test 2.2	100	2.956	93
Test 2.3	295	2.245	549
Test 2.4	1777	1.764	5546
Test 2.5	2007	1.344	7487
Problem 3			
Test 3.2	142	0.000	656
Test 3.3	620	0.000	7730
Test 3.4	5776	0.000	405316
Problem 4			
Test 4.2	189	0.000	2252
Test 4.3	1141	0.000	65298

The number of columns generated while solving the linear relaxation of the problem (root of the branch-and-bound tree) varies from 300 (Test 1.2) to 10169 (Test 3.4). The columns are generated at each branching node, their total number being as high as 525593 for Test 3.4. As can be seen from Table 5, in all test problems the number of branching nodes needed to obtain the integer solution was significant. In fact, problems of this type are very difficult to solve, not only because of their combinatorial nature which influences the solution time spent in the subproblems, but also because of the large number of the constraints in the master problem. Among these constraints, the most difficult to satisfy are the same flight departure time constraints. In fact, as the number of these constraints increases according to increase in the crew size, it becomes more and more difficult and time consuming to solve the corresponding master problem. Also, the number of fractional variables in the linear relaxation solution is large. It ranges from 8 (Test 1.2) to 390 (Test 2.5). Note that an integrality gap exists in the majority of tested problems, one as high as 3% in Test 2.2. The value of the lower bound of the integer solution rapidly increases when imposing decisions on time related variables. In fact, we observed that the value of the lower bound is equal to the value of the optimal integer solution when it becomes impossible to impose further decisions on the time variables.

Thus, we continue branching on the flow variables by using the value of the optimal integer solution as the lower bound. However, to obtain the optimal integer solution much further work must be done due to the very large number of fractional flow variables.

The computing times presented in Table 5 can be further reduced by applying a heuristic branching on flow variables, which involves simultaneously fixing several fractional flow variables to either one or zero. For each node of the branching tree only one branch is created and explored without any backtracking. The corresponding results obtained by using this heuristic branching strategy are presented in Table 6.

As Table 6 shows, the computing time is significantly reduced for all test problems without influencing the quality of the integer solution. In fact, the optimal integer solution is obtained for all tested problems. Thus, the optimality gap is equal to zero in all considered cases.

Table 6: Model 1: Computing times - heuristic branching on flow variables

Tests	BB Nodes	Optimality Gap (%)	CPU (sec)
Problem 1			
Test 1.2	11	0	1
Test 1.3	14	0	3
Test 1.4	9	0	5
Test 1.5	17	0	12
Test 1.6	25	0	21
Test 1.7	15	0	33
Problem 2			
Test 2.2	43	0	62
Test 2.3	68	0	315
Test 2.4	136	0	1957
Test 2.5	51	0	1237
Problem 3			
Test 3.2	13	0	147
Test 3.3	51	0	769
Test 3.4	53	0	4369
Problem 4			
Test 4.2	75	0	943
Test 4.3	61	0	5105

Having in mind the complexity and size of the considered problems, we can conclude that the computational times presented in Table 6 are very good. They show that our solution approach is very well suited to commercial-size problems. For the smaller problems, occurring the most frequently in practice, such as Test 1.2 - Test 1.7, the computing time ranges from 0.8s to 32.7s. Obviously, as the crew size increases, the total computing time increases significantly. For example, the large problem represented by Test 2.5 is solved in only 20 minutes. The huge problems, such as Test 3.4 and Test 4.3, are solved, respectively, in 1 hour 13 minutes and 1 hour 25 minutes. It must be remembered that these two problems correspond to high volume disturbances, e.g. snow storm, which happen very rarely and are usually announced several hours in advance, leaving enough time to solve potential scenarios.

3.2. Model 2

The branching method used in Model 1 was not able to efficiently solve the problem. For example, the optimal integer solution for the small problem Test 1.2 was found after exploring 3642 branching nodes, which took 97.8 seconds. The corresponding branch-and-bound tree was completely explored in order to determine the optimal integer solution. With the same branching strategy, Test 1.3 was unsolved after 1 hour of CPU time and 63479 explored nodes, when we stopped the process.

By examining the branching tree we found out that, among the fractional flow variables in the solution of the linear relaxation of the problem, there were the fractional flows associated with the artificial crew. It follows that Model 2 is even more difficult than Model 1 because of the large difference between the solutions obtained when a fractional flow associated with the artificial crew is fixed to 1 (implying penalties for a flight uncovering) and to 0 (no penalties).

Since the lower bound of the optimal integer solution does not change once the flight departure times are fixed, every branch must be explored to the very end, when the process backtracks to another node and continues the exploration. To reduce the computing time, a good lower bound must be provided during the branching process. In order to achieve this objective, we introduced a new type of decision based on the flow variables associated with the artificial crew. Hence, our new branching strategy permits us to impose branching decisions first on flow variables corresponding to the artificial crew, and then on time related variables and on remaining flow variables, if necessary. The new steps of the branching strategy can be summarized as follows:

Step 1: Choose a decision variable and define two possible branching decisions. When a fractional solution is obtained after solving given branching node i, first identify, if any, fractional flow variables associated with the artificial crew. Find among them the one with the largest value. Choose this variable for the next branching decision. Create two branches by fixing the value of the chosen variable to 0 and to 1.

Step 2: Identify the most promising decision. The most promising branching decision is that corresponding to value 0 of the chosen fractional variable. By fixing a fractional flow associated with the artificial crew to 0, we force the corresponding flight to be completely covered by real crew members. However, if such a solution does not exist, the corresponding branch will be immediately

cut and the search will continue by correctly fixing the given flow variable to 1.

Step 3: Impose the selected decision. Impose the current branching decision in the master problem by eliminating all columns which do not satisfy it. Also impose it in the subproblem by removing arcs not corresponding to the imposed decision.

Step 4: Solve the current branching node. Reoptimize the resulting master problem by column generation. All columns generated at this node are consistent with the last imposed decision.

In the case where some flow variables in the master problem are still fractional, but there are no fractional variables associated with the artificial crew, the search for the optimal integer solution continues by applying the branching scheme used in Model 1 and described in Section 1.2. Results obtained by applying Model 2, which requires us either to cover a flight by a complete crew, or to leave it uncovered, are presented in

Table 7. The table reveals that the results obtained with Model 2 are still far better than those presented in Table 3. Similarly to Model 1, the number of uncovered tasks is reduced, on average, by 64% with respect to the solution presented in Table 3, while the number of misplaced crew members in the optimal solution increased, on average, by 17%. Although the number of misplaced crew members is higher than in the initial situation, once again the great majority of most urgent problems, being those concerning the considered day of operations, have been successfully solved.

Table 7: Model 2: Solution with crew schedule reoptimization

Tests	Uncovered Flights	Misplaced Crew Members	Delayed Flights	Average Delay (min)
Problem 1				
Test 1.2	6	8	4	46
Test 1.3	6	11	4	46
Test 1.4	6	16	4	46
Test 1.5	6	19	4	46
Test 1.6	6	24	4	46
Test 1.7	6	27	4	46
Problem 2				
Test 2.2	4	7	7	42
Test 2.3	4	9	9	35
Test 2.4	4	11	9	35
Test 2.5	4	13	9	35
Problem 3				
Test 3.2	4	10	14	34
Test 3.3	4	15	14	34
Test 3.4	4	20	14	34
Problem 4				
Test 4.2	7	14	21	32
Test 4.3	7	21	21	32

While the number of resources, nodes and commodities remains the same as in Model 1, the number of arcs is slightly smaller than in Model 1. This reduction is due to the lower number of arcs associated with the artificial commodity. The number of master problem constraints is identical to that for Model 1. The computational results obtained by using the new branching technique described previously are presented in Table 8. All problems are solved to optimality. The columns of the table correspond to the columns of Table 5. As can be seen from Table 8, the integrality gap is very large in all but the last five tests. To find an optimal integer solution, it was necessary to explore more branching nodes than in Model 1. Thus, this last model is more difficult to solve than the previous one and the corresponding computing times are larger. Once again, CPU times are very good for smaller and medium-size problems and increase significantly for large problems.

Table 8: Model 2: Computing times - new optimal branching method

Tests	BB Nodes	Integrality Gap (%)	CPU (sec)
Problem 1			
Test 1.2	13	18.827	1
Test 1.3	19	12.053	4
Test 1.4	48	18.944	14
Test 1.5	58	14.760	28
Test 1.6	204	18.984	110
Test 1.7	2050	15.959	1624
Problem 2			
Test 2.2	276	13.936	346
Test 2.3	424	15.546	1790
Test 2.4	882	16.318	7565
Test 2.5	1520	16.694	25294
Problem 3			
Test 3.2	252	0.000	1199
Test 3.3	675	0.000	10615
Test 3.4	4219	0.000	213468
Problem 4			
Test 4.2	238	0.000	2847
Test 4.3	1293	0.000	93089

Table 9 shows the efficiency of the new branching technique in comparison to the original one used in Model 1. For the original branching method, we restricted the maximum CPU time to 1 hour. The second column of the table gives the number of branching nodes explored before the process was completed (CPU < 1 hour) or was stopped (CPU = 1 hour). The third column gives the percentage increase of the lower bound over the optimal solution of the linear relaxation of the problem. The fourth column gives the residual gap in percentage points. The fifth column shows the CPU time in seconds before the process was stopped or the corresponding computing time before the last value of the lower bound was reached. The next four columns of the table, which refer to the solution obtained when applying the new branching method, correspond to the columns of the original branching method.

Table 9: Model 2: Comparison of the two branching methods

	Original Branching Method				New Branching Method			
Tests	B&B Nodes	LB Increase (%)	Residual Gap (%)	CPU (sec)	B&B Nodes	LB Increase (%)	Residual Gap (%)	CPU (sec)
Test 1.2	3642	18.819	0.008	97.8	10	18.827	0.000	0.8
Test 1.3	63479	6.926	5.127	3600.0	14	12.053	0.000	2.8
Test 1.4	38719	0.023	18.921	3600.0	10	18.944	0.000	5.8
Test 1.5	22452	0.021	14.739	3600.0	10	14.760	0.000	12.1
Test 1.6	15360	0.015	18.969	3600.0	8	18.984	0.000	21.0
Test 1.7	10117	0.015	15.944	3600.0	12	15.959	0.000	48.4

Table 9 reveals that the original branching technique is completely inefficient for Model 2, where the integrality gap between the optimal fractional solution of the problem and the optimal integer solution is very large. Only one of the problems was solved in under 1 hour. Moreover, the value of the lower bound on the last explored node (before the process was stopped) was very far from its final value, in all except the smallest test, Test 1.2, where the residual gap was 0.008 %.

For the other tests, even after 1 hour CPU time, the residual gap is significant ranging from 5.127 % to 18.969 %. The number of nodes explored within 1 hour when using the branching technique is very large, too. In contrast, the new branching method is very efficient for all test problems. A maximum of 14 nodes have been explored before the final value of the lower bound was established and it took only 48.4s to increase the lower bound to its final value for Test 1.7, the most difficult of the test problems presented in Table 9. Furthermore, the value of the last lower bound found by the new branching method is equal to the value of the optimal integer solution for all test problems presented in the table. We can conclude that, without using the new branching technique, Model 2 could not be used to solve even very small problems.

The heuristic branching strategy on flow variables, which was applied when solving test problems with Model 1, is applied also when solving test problems with Model 2. By applying the heuristic branching strategy, instead of the optimal one, the computing times presented in Table 8 were further improved, while the corresponding solutions remained optimal. The results are presented in Table 10.

Table 10: Model 2: Computing times - heuristic branching on flow variables

Tests	BB Nodes	Optimality Gap (%)	CPU (sec)
Problem 1			
Test 1.2	12	0	1
Test 1.3	20	0	3
Test 1.4	20	0	9
Test 1.5	24	0	16
Test 1.6	18	0	27
Test 1.7	24	0	58
Problem 2			
Test 2.2	254	0	328
Test 2.3	264	0	1518
Test 2.4	316	0	5618
Test 2.5	431	0	18938
Problem 3			
Test 3.2	36	0	263
Test 3.3	75	0	1607
Test 3.4	77	0	6553
Problem 4			
Test 4.2	52	0	867
Test 4.3	87	0	8663

3.3. Comparison between Model 1 and Model 2

Table 11 was created in order to compare the optimal solutions of the two models. It shows the total number of uncovered flights corresponding to uncovered tasks, for both models. Flights covered by an incomplete crew are referred to as partially covered flights, while the others are referred to as completely uncovered flights. The total number of uncovered flights for Model 1 represents the sum of the completely and partially uncovered flights.

Table 11: Residual uncovered flights for the two implemented models

	Uncovered Flights			
Tests	Model 1			Model 2
	Completely	Partially	Total	Total
Problem 1				
Test 1.2	4	2	6	6
Test 1.3	4	2	6	6
Test 1.4	4	2	6	6
Test 1.5	4	2	6	6
Test 1.6	4	2	6	6
Test 1.7	4	2	6	6
Problem 2				
Test 2.2	1	5	6	4
Test 2.3	1	5	6	4
Test 2.4	1	6	7	4
Test 2.5	1	6	7	4
Problem 3				
Test 3.2	3	2	5	4
Test 3.3	3	3	6	4
Test 3.4	3	2	5	4
Problem 4				
Test 4.2	7	0	7	7
Test 4.3	6	2	8	7

The results presented in Table 11 show that both models leave the same number of uncovered flights for all the tests corresponding to Problem 1 and Test 4.2. For the other tests, Model 2 leaves fewer uncovered flights than Model 1. This is due to the fact that those crew members who have been assigned to the partially covered flights in the solution produced by Model 1 are used by Model 2 to completely cover some other flights.

3.4. Comparison between the New and the Traditional Approach to the Problem

This last part of the section is devoted to demonstrating the advantages of our new optimization approach compared to the traditional sequential approach where the flight schedule is fixed first and then input to the operational multi-crew scheduling problem. All the flexible scheduled flights from Table 1 were first converted to the fixed scheduled flights and then all the test problems were solved.

Table 12 shows the number of residual problems that each approach left. In both Model 1 and 2, the new approach left far less uncovered flights than the traditional approach. The ratio of uncovered tasks left by the traditional approach compared to uncovered tasks left by Model 1 ranges from 1.94 to 3.29, the average being 2.52. The average number of misplaced crew members in the traditional approach is 1.01 times higher compared to Model 1. The corresponding minimum and maximum values are 0.70 and 1.40, respectively.

Table 12: Residual problems of the simultaneous and the sequential approach

	Model 1				Model 2			
Tests	Uncovered Tasks		Misplaced Members		Uncovered Tasks		Misplaced Members	
	Schedule		Schedule		Schedule		Schedule	
Problem 1	Flexible	Fixed	Flexible	Fixed	Flexible	Fixed	Flexible	Fixed
Test 1.2	10	20	7	5	6	11	8	6
Test 1.3	16	31	10	7	6	11	11	8
Test 1.4	20	40	14	11	6	11	16	12
Test 1.5	26	51	17	12	6	11	19	14
Test 1.6	30	60	21	15	6	11	24	18
Test 1.7	36	71	24	17	6	11	27	20
Problem 2								
Test 2.2	7	18	7	7	4	12	7	10
Test 2.3	10	26	10	10	4	12	9	13
Test 2.4	13	34	13	13	4	12	11	16
Test 2.5	16	42	16	16	4	12	13	19
Problem 3								
Test 3.2	8	30	10	14	4	15	10	14
Test 3.3	12	45	15	21	4	15	15	21
Test 3.4	16	60	20	28	4	15	20	28
Problem 4								
Test 4.2	14	46	14	18	7	23	14	18
Test 4.3	21	69	21	27	7	23	21	27

Similar results have been obtained for Model 2. The traditional approach leaves on average 2.59 times more uncovered flights than the new simultaneous approach, the minimum value being 1.83 and the maximum 3.29. Meanwhile, the number of crew members misplaced by the traditional approach is, on average, 1.08 times greater compared to the new approach, the corresponding smallest and largest values being, respectively, 0.73 and 1.46.

We can conclude that the new approach, that simultaneously solves the operational flight and multi-crew scheduling problem, produces solutions whose quality is far better than that of the traditional approach with respect to both the number of uncovered flights and the number of misplaced crew members. Due to the increased complexity of the problem, the solution process of the new approach is slower than that of the traditional approach (12.44 times for Model 1 and 9.54 times for Model 2). However, computational times of our new approach are still very good for small and medium volume problems which the airline companies most commonly face.

4. CONCLUSION AND EXTENSIONS

In this paper we have studied the operational multi-crew scheduling problem with interchangeable positions, flexible flight departure times and fixed aircraft itineraries. Two original models were developed and successfully solved. The objectives were to minimize the total number of uncovered tasks (Model 1) or the total number of uncovered flights (Model 2), as well as the total delay of all flights and the number of crew members whose planned activities for the next day of operations must be consequently changed. The problem has been modeled as an integer nonlinear multi-commodity network flow problem with time windows and additional flight precedence and same flight departure time constraints. The problem is very difficult to solve due to the numerous constraints in the master problem, the same flight departure time constraints being the most difficult to satisfy. Moreover, an integrality gap usually exists and can be very large (up to 19% for the set of problems tested with Model 2). We used a specialized branch-and-bound algorithm, based on the values of the time variables and of the flow variables between pairs of tasks, for each of the proposed models. The branching method for each model exploits the particular characteristics of the problem in order to increase the lower bound of the integer solution as the branching process proceeds. It follows that we do not need to explore the whole branch-and-bound tree.

The proposed solution approaches are the very first models involving simultaneous crew and flight scheduling for multi-crew. They are tested, along with the new branching technique, on several input data sets. The reported results are good from the point of view of solution quality. The great majority of most urgent problems, being those concerning the considered day of operations, have been successfully solved, leaving some problems for the next day of operations. The reported computing times, which are very good for small and medium problems (less than a minute for Problem 1), show that our approach is very efficient for the most common commercial problems. Although the reported solution time is higher for larger problems, it can be cut by an important factor by using more powerful computers available today or in the near future. Another possibility is to use aggressive branching strategies. While this is appropriate for commercial products, here we prefer using less aggressive branching strategies that produce solutions of better quality. Finally, the smaller time flexibility would help in reducing the time needed to solve a problem. As it can be seen from Stojković and Soumis (2001), problems with fixed scheduled flights only are solved in few seconds. Even if the computing time grows with the number of required crew members per crew, it remains reasonable when crew members are grouped into few sub-teams.

The two optimization approaches proposed in this paper could be combined into a single approach which would permit us to keep in the optimal solution some partially covered flights under the condition that the number of crew members assigned to any of these flights is not smaller than the minimum fixed in advance. Otherwise, a partially covered flight must be left uncovered. Thus the present research represents the basis for future developments. Finally, both the economical aspects and the robustness of a proposed solution could be improved by introducing a more complex cost function.

Acknowledgments: This work was supported by the Quebec Government (Programme Synergie du Fonds de Développement Technologique) and by the Natural Sciences and Engineering Council of Canada.

REFERENCES

[1] Carpaneto, G., and Toth, P., "Some new branching and bounding criteria for the asymmetric traveling salesman problem", *Management Science*, 26 (1980) 736-743 .

[2] Desaulniers, G., Desrosiers, J., Ioachim, I., Soumis, F., and Solomon, M.M., "A unified framework for time constrained vehicle routing and crew scheduling problems", Les Cahiers du GERAD G-94-46, École des Hautes Études Commerciales, Montréal, Canada, 1994, Revised 1998.

[3] Desrosiers, J., Soumis, F., and Sauvé, M., "Lagrangian relaxation for routing with time windows", in: J.P.Brans (ed.), *Operational Research*, 1984.

[4] Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F., "Time constrained routing and scheduling", *Handbooks in Operations Research and Management Science*, Volume on Networks, North Holland, Amsterdam, 1994.

[5] Gélinas, S., Desrochers, M., Desrosiers, J., and Solomon, M.M., "A new branching strategy for time constrained routing problems with application to backhauling", *Annals of Operations Research*, 61 (1995) 91-109 .

[6] Ioachim, I., Desrosiers, J., Soumis, F., and Bélanger, N., "Fleet assignment and routing with schedule synchronization constraints", Les Cahiers du GERAD G-94-48, École des Hautes Études Commerciales, Montréal, Canada, 1994, Revised 1997.

[7] Ioachim, I., Gélinas, S., Soumis, F., and Desrosiers, J., "A dynamic programming algorithm for the shortest path problem with time windows and linear node costs", *Networks*, 31 (1998) 193-204.

[8] Lettovský, L., "Airline recovery: An optimization approach", Ph.D. dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1997.

[9] Lettovský, L., Johnson, E.L., and Nemhauser, G.L., "Airline crew recovery", *Transportation Science*, 34 (4) (2000) 337-348.

[10] Luo, S., and Yu, G., "Airline schedule perturbation problem: Landing and takeoff with nonsplitable resource for the ground delay program", in: Gang Yu (ed.), *Operations Research in the Airline Industry*, Kluwer Academic Publishers, Norwell, Massachusetts, 1998a, 404-432.

[11] Luo, S., and Yu, G., "Airline schedule perturbation problem: Ground delay program with splitable resources", Gang Yu (ed.), *Operations Research in the Airline Industry*, Kluwer Academic Publishers, Norwell, Massachusetts, 1998b, 433-460.

[12] Monroe, W., and Chu, H.D., "Real-time crew rescheduling", *INFORMS Conference*, New Orleans, 1995.

[13] Stojković, M., Soumis, F., and Desrosiers, J., "The operational airline crew scheduling problem", *Transportation Science*, 32 (3) (1998) 232-245.

[14] Stojković, M, and Soumis, F., "An optimization model for the simultaneous operational flight and pilot scheduling problem", *Management Science*, 47 (2001) 1290-1305.

[15] Wei, G., Yu, G., and Song, M., "Optimization model and algorithm for crew management during airline irregular operations", *Journal of Combinatorial Optimization*, 1 (1997) 305-321.