**An Exact Solution
Approach for the
PBS Problem**

H. Achour, M. Gamache,
F. Soumis, G. Desaulniers

An Exact Solution Approach for the PBS Problem

Heykel Achour

Michel Gamache

François Soumis

Guy Desaulniers

GERAD and École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
{heykel.achour;francois.soumis;guy.desaulniers}@gerad.ca
michel.gamache@polymtl.ca

December 2004

Abstract

This paper introduces the first exact approach for constructing aircrew member personalized monthly work schedules when a preferential bidding system (PBS) is used. With such a system, each employee bids for their preferred activities, yielding a bidding score for each feasible schedule. The PBS problem thus consists of assigning to each employee a schedule that maximizes, in order of seniority, its preferences while covering all crew pairings. The proposed exact solution approach relies on column generation and, when a tentative maximum score for a crew member is established, it explicitly enumerates for that employee all feasible schedules with that score. Tests on real-life cases show that this approach can substantially improve the quality of the solutions produced by the best known existing method (Gamache et al., 1998b) in similar solution times.

Résumé

Dans cet article nous proposons une approche exacte de résolution du problème de fabrication d'horaires personnalisés avec priorités. À notre connaissance, aucune méthode optimale n'a été rapportée dans la littérature pour résoudre ce problème. La difficulté de l'obtention d'une solution optimale est montrée à travers la discussion de la résolution de deux formulations alternatives. On présente donc l'utilité du choix d'une méthode séquentielle satisfaisant tous les critères d'optimalités. La non fixation des horaires des pilotes lors de la résolution des problèmes par la technique de génération de colonnes intégrée dans un algorithme de branchement est un élément déterminant pour assurer l'optimalité de cette méthode. On propose un nouveau schéma de branchement pour permettre le choix d'un horaire parmi plusieurs de pointage quivalent pour un employ senior dans la mesure où cet horaire permet de trouver un meilleur horaire à l'employé possédant le moins d'ancienneté. Des tests sur des instances réelles des préférences des pilotes d'Air Canada ont été effectués et leurs horaires ont été considérablement améliorés.

1   Introduction

In the airline industry, the problem of constructing personalized monthly work schedules consists of designing a monthly schedule for each employee while taking into account their preferences and a list of preassigned activities such as vacations and training periods. The set of constructed schedules must cover a set of previously determined crew pairings. Usually, integer values are assigned to the activities (pairings, rest periods, ...) in order to reflect the preferences of each employee. The score of each schedule for each employee is computed as the sum of the employee values of the activities it contains. Two modes can be used to build these schedules. In the first mode, the objective consists of maximizing the global satisfaction of the employees, that is, the overall sum of the selected schedule scores. Typically, the set of activities only includes the pairings in this case. This mode is known as the rostering problem and is mainly used in large European airlines such as Air France (Giafferri et al., 1982, Gontier, 1985, Gamache *et al.*, 1998a), Alitalia (Nicoletti, 1975, Marchettini, 1980, Sarra, 1988, Lufthansa (Glanert, 1984), and SwissAir (Tingley, 1979). This mode has also been used at Air New-Zealand (Ryan, 1992) and El-Al Israel Airline (Mayer, 1980).

The second mode is similar to the first one except that the choice of preferences is more elaborate and the assignment of these preferences must be done in accordance with the seniority of the employees. Employees can formulate preferences based on the destinations, the pairing departure times, the pairing durations, the number of weekends off, etc. This mode of constructing schedules with priorities is most often used in large North American carriers and is known as the preferential bidding system (PBS).

The PBS problem consists of constructing the maximum-score feasible schedule for each employee according to his seniority order. A schedule is said to be feasible if it respects constraints imposed by the security rules in the airline industry, the collective agreement between the employees and the company, and the activities preassigned to the employees. Moreover, the set of computed schedules must cover the set of pairings planned during the month. Due to the complexity of the PBS problem, no exact approaches have yet been proposed in the literature for solving it. This paper fulfills this gap by proposing the first exact solution method. It also reports computational results on real-world problems which show that this new method can substantially improve the quality of the solutions produced by the best known existing heuristic method, that of Gamache et al. (1998b), in similar computational times.

This paper is organized as follows. In Section 2, we describe the problematic raised by the PBS problem and briefly review the heuristic approaches proposed in the literature to deal with this problem. In Section 3, we present the new exact method. Test results are reported in Section 4, while conclusions are drawn in Section 5.

2   Problematic and literature review

When solving the PBS problem, one has to optimize m different objectives (one for each of the m employees) that must be treated in lexicographic order from the objective of the most

senior employee, numbered $k = 1$, to the objective of the most junior employee, numbered $k = m$. This order implies that a tiny increase in the value of the kth objective must be prioritized over any other increase in any subsequent objective $l > k$. Sherali and Soyster (1983) proposed an approach that consists of finding a set of weights $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ to reflect the priority of the different objectives. A first model, denoted M^1, for the PBS problem based on the Sherali and Soyster's approach is:

$$(M^1) \begin{cases} Max\ Z = \sum_{i=1}^{m} \left(\lambda_i \sum_{j \in \Omega_i} c_{ij} x_{ij} \right) & (1) \\[2ex] \text{subject to} \\[1ex] \sum_{i=1}^{m} \sum_{j \in \Omega_i} a_{ijp} x_{ij} = b_p & p = 1, \ldots, P & (2) \\[2ex] \sum_{j \in \Omega_i} x_{ij} = 1 & i = 1, \ldots, m & (3) \\[2ex] x_{ij} \in \{0, 1\} & i = 1, \ldots, m,\ \ \forall j \in \Omega_i & (4) \end{cases}$$

In this model, Ω_i is the set of feasible schedules for employee i; c_{ij} represents the score of schedule j for employee i; x_{ij} is a binary variable that takes the value 1 if schedule j is chosen for employee i, and 0 otherwise; P is the set of pairings to cover; a_{ijp} is a parameter equal to 1 if schedule j for employee i contains pairing p, and 0 otherwise and b_p represents the number of employees required by pairing p.

In M^1, the objective function (1) aims at maximizing the weighted sum of the scores of the employee schedules, where the λ_i prioritize the scores of the most senior employees. The constraint set (2) ensures that the set of selected schedules covers all the pairings with the appropriate number of employees. Constraints (3) guarantee that a schedule is built for each employee. Binary requirements on the x_{ij} variables are given by (4).

The schedule scores, c_{ij}, are positive or negative numbers that may necessitate 32 bits to write. According to Sherali and Soyster (see Theorem 2.1 in their paper), the weights λ_i can be determined by first computing the following two constants for each employee i: ν_i and α_i, the largest and the smallest positive difference between the scores of two feasible schedules for employee i, respectively. Then, these values are used to define two other constants:

$$\begin{aligned} \nu &= \max \{\nu_i\ :\ i = 1, \ldots, m\}, \\ \alpha &= \min \{\alpha_i\ :\ i = 1, \ldots, m\}. \end{aligned}$$

Finally, to take into account the seniority order, the weights are set to $\lambda_i = (\nu/\alpha)^{m-i}$ for $i = 1, \ldots, m$. In the PBS problem, ν may be as large as 2^{32} while α is often equal to 1. Therefore, this approach cannot be used in practice because these weights are much too large for any computer, even for small values of m.

Following this observation, it becomes obvious that schedules must be built sequentially, that is, one after the other, from the most senior employee to the most junior one. In such a sequential approach, a problem is defined for each employee. It consists of constructing the best schedule for this employee and involves an objective function defined only with the employee preferences. While looking for the best schedule of the current employee, the method must ensure that the senior employees (those that are more senior than the current employee) are also assigned to a schedule that maximizes their score, the junior employees (those that are more junior than the current employee) will receive a feasible schedule, and that it will be possible to cover all pairings. Moreover, when more than one schedule with the highest score can be built for the current employee, the method must choose the one schedule that has the smallest impact on the scores of the schedules of the junior employees. In summary, to be exact, a sequential approach must construct for each employee a schedule that:

1. is feasible;
2. ensures that the set of residual pairings (those remaining to cover after choosing the schedule of the current employee) can be covered by a set of feasible schedules for the junior employees;
3. is a maximum score schedule among the set of schedules satisfying th two preceding criteria;
4. among the set of schedules satisfying the three preceding criteria, allows building the best schedules for the junior employees.

A sequential approach that fulfills these four exactness criteria is as follows. For $k = 1, \ldots, m$, solve $M^2(k)$, the following model:

$$
(M^2(k))
\begin{cases}
Max \ Z = \displaystyle\sum_{j \in \Omega_k} c_{kj} x_{kj} & (5) \\[2ex]
\text{subject to} & \\[1ex]
\displaystyle\sum_{i=1}^{m} \sum_{j \in \Omega_i} a_{ijp} x_{ij} = b_p \qquad p = 1, \ldots, P & (6) \\[2ex]
\displaystyle\sum_{j \in \Omega_i} c_{ij} x_{ij} = Z_{IP}^i \quad i = 1, \ldots, k-1 & (7) \\[2ex]
\displaystyle\sum_{j \in \Omega_i} x_{ij} = 1 \qquad i = 1, \ldots, m & (8) \\[2ex]
x_{ij} \in \{0,1\} \quad i = 1, \ldots, m, \ \ \forall j \in \Omega_i. & (9)
\end{cases}
$$

In this model, Z_{IP}^i represents the score of the best schedule for employee $i \in \{1, \ldots, k-1\}$ that has been identified when solving $M^2(i)$. The objective function of $M^2(k)$ maximizes only the score of the schedule of employee k while imposing through constraints (7) that the first $k - 1$ employees receive a schedule whose score is equal to the one found in the

previous iterations. Constraints (6), (8) and (9) are identical to constraints (2)–(4). Note that the equivalence between the sequence of problems $M^2(k)$ and the model M^1 is ensured by the Sherali and Soyster (1983) theorem.

For real-world cases, solving the sequence of problems $M^2(k)$ is also impractical since it involves finding an integer solution for m generalized set partitioning problems. In fact, to the best of our knowledge, no exact solution approaches for the PBS problem have been proposed in the literature. Most known methods use similar greedy heuristics. Quantas (Moore *at al.*, 1978), and CP Air (Byrne, 1988) developed their own method for solving the PBS problem. For each employee starting from the most senior, such a method builds its feasible schedule by iteratively selecting the activities with the highest scores among those not yet completely covered by the senior employees. During this construction phase, the method checks that there are enough junior employees available to cover the remaining pairings on each day of the month. When this is not the case for a given day, the current employee is forced to take a pairing on that day. When a schedule has been designed for each employee, a local search phase that swaps pairings between the schedules is usually applied to improve the schedule scores.

Gamache et al. (1998b) have proposed another sequential solution approach, which is, to our knowledge, the most efficient known heuristic for the PBS problem. It has the advantage of building for each employee a schedule that is considered optimal given the schedules previously assigned to the senior employees. It consists of solving for each employee, from the most senior to the most junior, an integer linear program whose objective is to maximize the score of the current employee schedule while taking into account the schedules built at the previous iterations. This program also ensures that a feasible schedule can be built for all remaining junior employees and that all pairings can be covered. The computed solution for this problem provides a schedule for the current employee, which is fixed before moving on to the construction of the next employee schedule. The integer program, denoted $M^3(k)$, solved for employee k is:

$$(M^3(k)) \begin{cases} Max \ Z = \sum_{j \in \Omega_k^k} c_{kj} x_{kj} & (10) \\[2mm] \text{subject to} \\[2mm] \sum_{i=1}^{k-1} a_{ij_ip} + \sum_{i=k}^{m} \sum_{j \in \Omega_i^k} a_{ijp} x_{ij} \ = \ b_p \qquad p = 1, \dots, P & (11) \\[2mm] \sum_{j \in \Omega_i^k} x_{ij} \ = \ 1 \qquad i = k, \dots, m & (12) \\[2mm] x_{ij} \ \in \ \{0,1\} \quad i = k, \dots, m, \ \ \forall j \in \Omega_i^k & (13) \end{cases}$$

where j_i is the index of the schedule fixed for the employee $i \in \{1, \dots, k-1\}$ and Ω_i^k represents the set of feasible schedules for the employee $i \in \{k, \dots, m\}$ that are still available

at iteration k (i.e., they do not contain any pairing fully covered by the fixed schedules of the previous employees). To obtain acceptable solution times, Gamache et al. (1998b) proposed to solve each model $M^3(k)$ using a branch-and-price approach, that is, a column generation method embedded in a branch-and-bound search tree (see Barnhart et al., 1998, and Desaulniers et al., 1998). For a given k, a column generation subproblem is defined for each of the remaining employees, i.e., employees $i \in \{k, \ldots, m\}$. Each subproblem corresponds to a longest path problem with resource constraints (see Irnich and Desaulniers, 2004) on an acyclic network G_i^k specific to the corresponding employee i, and is solved by a generalization of the dynamic programming algorithm of Desrochers and Soumis (1988). The index k indicates that the network changes with k since all arcs representing a pairing are removed when the corresponding pairing becomes completely covered by the fixed schedules of the senior employees. The resource constraints are used to model various security and collective agreement rules that restrict the legality of a schedule (see Gamache et al., 1998b). Examples of the resources considered are the number of flight credits, the number of consecutive working days, and the number of days off per month. The first two resources must not exceed prescribed values while the last must be at least equal to another prescribed value. Each path in G_i^k that satisfies all resource constraints corresponds to a feasible schedule for employee i. Furthermore, all feasible schedules in the set Ω_i^k of available schedules for this employee are represented by a resource-feasible path in this network.

In order to speed up the overall solution process, Gamache et al. (1998b) proposed to solve a relaxed version of $M^3(k)$ where the constraints (13) for $i = k+1, \ldots, m$ are replaced by their continuous relaxations. However, after fixing the schedule of several employees, this strategy sometimes leads to an infeasible relaxed problem $M^3(k)$. In this case, backtracking is performed, that is, the (non relaxed) models $M^3(i)$ for $i = k-1, k-2, \ldots, \ell$ are solved in this reverse order until finding a feasible problem $M^3(\ell)$. The newly computed schedule for the employee ℓ is then fixed and the solution process returns to a forward mode by solving a newly defined relaxed problem $M^3(\ell + 1)$.

This solution approach is heuristic since it does not allow exchanging the best-score schedules of the senior employees when solving the problem for a junior employee, that is, it does not satisfy the last of the four exactness criteria mentioned above. Indeed, the existence of several schedules with the same score is quite frequent in real-world instances because of the diversity of the preferences offered to the employees and the number of possible substitutions. One way to improve the overall solution could be to exchange pairings between the schedules in a post-optimization phase as in Ryan et al. (1997). However, this would not guarantee optimality. It would be preferable to perform such exchanges durin the optimization process.

The next section introduces the first exact solution approach for the PBS problem. This approach is based on that of Gamache *et al.* (1998b). Therefore, in order to lighten the text, we will not provide all the details for the parts that are common to both approaches. The interested reader should also consult the paper of Gamache et al. (1998b).

3 An exact solution approach

The proposed exact solution approach is also a sequential approach in th sense that it solves a sequence of integer programs, one for each employee, from the most senior to the most junior employee, and fixes as the solution process progresses, the schedules of the senior employees. However, contrary to the heuristic method of Gamache et al. (1998b), these schedules are not necessarily fixed in the seniority order (although there is a high probability that this order will be followed) and, at each iteration, there might be more than one schedule fixed or none at all. Recall that, to be exact, this approach must satisfy the four criteri listed in the previous section. Therefore, since selecting prematurely and arbitrarily the schedule of a senior employee (among those with the same best score) may reduce the best score of a junior employee, the new approach postpones this selection until we can prove that there is a unique best-score schedule for the senior employee that can yield the optimal scores for all junior employees. To do so, the solution method includes a procedure that enumerates all the best-score schedules for each employee and applies different techniques for gradually discarding all these schedules but one. Note that this approach also backtracks when needed.

Throughout the solution process, the approach identifies certain pairings that must be part of the schedules of some senior employees so that they can attain their best scores. When these pairings are fully covered by these employees, they cannot be included in the other employee schedules. In the following we use the expression *residual schedules* to refer to the schedules for the other employees that do not contain these pairings.

This section describes the sequence of integer programs to be solved. Then, it discusses how this overall process can be sped up by using two relaxations of these programs. Finally, it explains how to enumerate the set of all best-score schedules for each employee and how this set can be reduced throughout the solution process to obtain a unique schedule.

3.1 The sequence of integer programs $IP(k)$

The exactness of the proposed method resides in the solution of a sequence of integer programs, denoted $IP(k)$, $k = 1, 2, .., m$. Each of these problems consists of finding a set of m feasible schedules, i.e., one for each employee, such that the score of the schedule assigned to the kth employee is maximized while optimal schedules are assigned to the senior employees. The problem $IP(k)$ is formulated as follows:

$$(IP(k)) \begin{cases} Max \ Z = \sum_{j \in \Omega_k^k} c_{kj} x_{kj} & (14) \\[2ex] \text{subject to} \\[2ex] \sum_{i=1}^{k-1} \sum_{j \in \widehat{\Omega}_i^k} a_{ijp} x_{ij} + \\ \sum_{i=k}^{m} \sum_{j \in \Omega_i^k} a_{ijp} x_{ij} = b_p & p = 1, \ldots, P & (15) \\[2ex] \sum_{j \in \widehat{\Omega}_i^k} x_{ij} = 1 & i = 1, \ldots, k-1 & (16) \\[2ex] \sum_{j \in \Omega_i^k} x_{ij} = 1 & i = k, \ldots, m & (17) \\[2ex] x_{ij} \in \{0,1\} & i = 1, \ldots, k-1, \ \forall j \in \widehat{\Omega}_i^k & (18) \\[2ex] x_{ij} \in \{0,1\} & i = k, \ldots, m, \ \forall j \in \Omega_i^k. & (19) \end{cases}$$

In this model, $\widehat{\Omega}_i^k$ denotes the set of residual schedules for employee $i \in \{1, \ldots, k-1\}$ that have the optimal score computed at iteration i, while Ω_i^k is the set of residual schedules for employee $i \in \{k, \ldots, m\}$. Model $IP(k)$ can be derived from model $M^2(k)$ by simply removing all schedules $j \in \Omega_i$, $i = 1, \ldots, k-1$, such that $c_{ij} \neq Z_{IP}^i$.

Note that the problems $IP(k)$ are all bounded. The following proposition relates the feasibility of these problems to the feasibility of the first one.

Proposition 1 *If problem $IP(1)$ is feasible, then all the problems $IP(k)$, $k \in \{2, \ldots, m\}$, are also feasible.*

Proof: Assume that $IP(k)$ is feasible for $k = 1$. Since it is bounded, it possesses at least one optimal solution. It is easy to see that this solution is feasible for the problem $IP(k+1)$, showing that $IP(2)$ is feasible. This argument can be repeated iteratively for $k = 2$ until $k = m - 1$. ∎

In practice, solving the sequence of problems $IP(k)$ (like the problems $M^2(k)$) requires a huge amount of time that is unacceptable. In order to substantially accelerate the solution time, our approach relies on a relaxation of $IP(k)$ and on a feasibility problem. In the relaxation of $IP(k)$, constraints (18) and constraints (19) for $i \neq k$ are replaced by non-negativity constraints. Given the scores computed for the senior employees, this relaxed version of $IP(k)$ provides an upper bound on the score of the best schedule for employee k. The feasibility problem is defined by replacing the constraints (19) in $IP(k+1)$ by non-negativity requirements and deleting the objective function. This problem is solved occasionally to verify that compatible best-score schedules can be found for the first k

employees. These two problems are denoted $MIP_{k \to k}(k)$ and $MIP(1 \to k)$, respectively, where $k_1 \to k_2$ indicates that integrality requirements are imposed on the variables associated with the employees from k_1 to k_2.

A summary of the solution approach is provided in Figure 1. An iteration begins by solving model $MIP(k \to k)$ to derive a tentative best-score S_k^* for employee k. Then the set $\widehat{\Omega}_k^k$ containing all residual schedules of score S_k^* for employee k is constructed and techniques reducing the sets $\widehat{\Omega}_i^k$ and Ω_i^k are applied. If a set of residual schedules $\widehat{\Omega}_i^k$ $i \in \{1, \ldots, k\}$, is reduced to a singleton (that is, there is a unique schedule that can be assigned to a senior employee), problem $MIP(1 \to k)$ is solved. If this problem is feasible, we continue with the next employee. Otherwise, the problem $MIP(k \to k)$ is altered by adding a cut restricting the score of the employee k schedule and solved again. As in Gamache et al. (1998b), this overall strategy may sometimes lead to an infeasible $MIP(k \to k)$ problem. In this case, backtracking is performed by solving the previous $IP(i)$ models, for $i = k - 1, k - 2, \ldots, \ell$, until finding a feasible problem $IP(\ell)$. Each component of the proposed approach are detailed in the following subsections.

3.2   Finding an upper bound S_k^*

The search for S_k^*, an upper bound on the score of the best schedule that can be assigned to employee k given the scores S_i^* already computed for the senior employees $i = 1, \ldots, k-1$, is done by solving the following model $MIP(k \to k)$:

$$(MIP(k \to k)) \begin{cases} Max\ Z = \sum_{j \in \Omega_k^k} c_{kj} x_{kj} & (20) \\ \text{subject to} & (21) \\ \sum_{i=1}^{k-1} \sum_{j \in \widehat{\Omega}_i^k} a_{ijp} x_{ij} & \\ + \sum_{i=k}^{m} \sum_{j \in \Omega_i^k} a_{ijp} x_{ij} = b_p \quad p = 1, \ldots, P & (22) \\ \sum_{j \in \widehat{\Omega}_i^k} x_{ij} = 1 \quad i = 1, \ldots, k-1 & (23) \\ \sum_{j \in \Omega_i^k} x_{ij} = 1 \quad i = k, \ldots, m & (24) \\ x_{ij} \geq 0 \quad i = 1, .., k-1, \forall j \in \widehat{\Omega}_i^k & (25) \\ x_{kj} \in \{0, 1\} \quad \forall j \in \Omega_k^k & (26) \\ x_{ij} \geq 0 \quad i = k+1, .., m, \forall j \in \Omega_i^k. & (27) \end{cases}$$
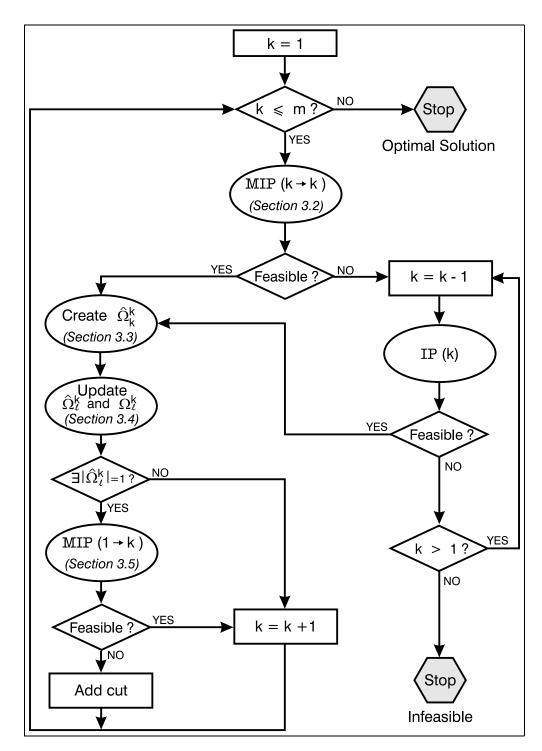
Figure 1: A diagram of the solution approach

Problem $MIP(k \to k)$ is solved by the exact branch-and-price method proposed by Gamache et al. (1998b), which will not be explained here. The existence of a feasible solution for this problem ensures that there exists a feasible schedule for employee k that can be completed by "fractional" feasible schedules for the other employees. The score of the computed schedule for employee k thus provides an upper bound on the best score. However, this upper bound is valuable only if it is possible to find an integer solution for all the employees. This is verified by solving the subsequent $MIP(k \to k)$ and $MIP(1 \to k)$ problems. It should be noted that $MIP(k \to k)$ is much easier to solve than $IP(k)$. Moreover, the schedules that can be obtained for the employees $i = k+1, \ldots, m$ by imposing integrality requirements on their associated variables would probably be useless (not optimal) because their preferences are not yet taken into account.

3.3   Creating $\widehat{\Omega}_k^k$

After solving a feasible $MIP(k \to k)$ whose optimal value is S_k^*, one must enumerate all feasible residual schedules for employee k that have a score of S_k^*. Two dynamic programming algorithms can be considered. Both are based on the generalization of the longest resource constrained path algorithm of Desrochers and Soumis (1988) (see Desaulniers et al., 1998), that is, the algorithm used to solve the column generation subproblem for employee k. Both use the network G_k^k. The first algorithm works with the cost of the arcs, the second one with their reduced cost (as in the column generation subproblem).

Given the network G_k^k and the value $\bar{S}_k$ of the maximum-cost feasible (resource-constrained) path between the source node $o(k)$ and the sink node $d(k)$, the first approach consists of solving a longest path problem with resource constraints while relaxing the dominance rule in order to accept all the partial paths that can lead to a solution with a cost of S_k^*. In the generalized version of the Desrochers and Soumis (1988) algorithm, any partial path between $o(k)$ and a node i is represented by a label $L_i = (Z_i, T_i)$, where Z_i is the cost of the partial path and T_i is an array indicating, for each resource, the quantity accumulated since node $o(k)$. Let $L_i^1 = (Z_i^1, T_i^1)$ and $L_i^2 = (Z_i^2, T_i^2)$ be two labels corresponding to two partial paths at a node i. The dominance rule used by Desrochers and Soumis can be described as follows: if $Z_i^1 \geq Z_i^2$ and $T_i^1 \leq T_i^2$ (i.e., each component of T_i^1 is less than or equal to the corresponding component of T_i^2), then the label L_i^2 can be eliminated since it is dominated by label L_i^1. Indeed, the second partial path cannot lead to an $o(k)$-to-$d(k)$ path that will be better than the one obtained by extending the first partial path. Since, in our case, we do not want only a path with the maximum cost, but all the paths having the score S_k^*, the dominance rule that we use is a relaxed one: eliminate L_i^2 if and only if $Z_i^1 > Z_i^2 + \bar{S}_k - S_k^*$ and $T_i^1 \leq T_i^2$. Thus, any partial path that can lead to an $o(k)$-to-$d(k)$ path with a cost of at least S_k^* is kept. Consequently, the first approach contains the following steps:

1. Compute $\bar{S}_k$ by solving the longest resource constrained path problem on G_k^k with the generalized algorithm of Desrochers and Soumis (1988).

2. Solve this problem again using this time the modified dominance rule to obtain a list of labels at node $d(k)$.

3. Among these labels, keep only those for which their cost component is equal to S_k^*. They represent all the feasible residual schedules for employee k with the maximum score S_k^*.

The second approach relies on the following assumption: the value of the linear relaxation of $MIP(k \rightarrow k)$ is S_k^*. When this assumption does not hold, one can solve again (after obtaining S_k^*) this relaxation with a cut added in the employee k subproblem that restricts all schedules generated for this employee to have a cost lower than or equal to S_k^* (see Gamache *et al.*, 1998b). This restriction forces the value of the linear relaxation to be equal to S_k^*. The result stated in the following proposition is the basis of the second approach.

Proposition 2 *Assume that $MIP(k \rightarrow k)$ is feasible and that its optimal value S_k^* is equal to that of its linear relaxation. Let $\tilde{X}_{LP}$ and $\tilde{\Pi}_{LP}$ be a pair of primal and dual optimal solutions for its linear relaxation, and consider any variable x_{kj} of cost S_k^* for which there exists a feasible solution $\tilde{X}_{MIP}$ to $MIP(k \rightarrow k)$ with $x_{kj} = 1$. The reduced cost $\bar{c}_{kj}$ of x_{kj} with respect to $\tilde{\Pi}_{LP}$ is zero.*

Proof: Two cases can occur depending on whether or not x_{kj} is a basic variable in $\tilde{X}_{LP}$. The first case (x_{kj} is basic) is trivial. Therefore, assume that x_{kj} is nonbasic in $\tilde{X}_{LP}$. By linear programming theory, we can write the cost z of any solution X as

$$z = S_k^* + \bar{c}_H X_H \tag{28}$$

where X_H is the vector of the components of X corresponding to the nonbasic variables in X_{LP}, and $\bar{c}_H$ is the vector of the reduced costs of these variables with respect to Π_{LP}. Now, notice that $\tilde{X}_{MIP}$ is a feasible solution of cost S_k^*. Therefore, setting $X = \tilde{X}_{MIP}$ in equation (28) yields

$$S_k^* = S_k^* + \bar{c}_H \tilde{X}_{MIP,H}$$

$$\Rightarrow \quad \bar{c}_H \tilde{X}_{MIP,H} = 0, \tag{29}$$

where $\tilde{X}_{MIP,H}$ is the vector of the components of $\tilde{X}_{MIP}$ corresponding to the nonbasic variables in $\tilde{X}_{LP}$.

Since $\bar{c}_H \leq 0$ ($\tilde{\Pi}_{LP}$ is optimal) and $\tilde{X}_{MIP,H} \geq 0$ ($\tilde{X}_{MIP}$ is feasible), we obtain that each term in the left-hand side of (29) is null. In particular, this applies to the term involving the variable x_{kj}, that is, $\bar{c}_{kj} x_{kj} = 0$. Finally, since we assumed that $x_{kj} = 1$ in $\tilde{X}_{MIP}$, we find that $\bar{c}_{kj} = 0$. ∎

This proposition shows that all feasible residual schedules for employee k with a score S_k^* correspond to paths in G_k^k with a null reduced cost. Furthermore, since the dual solution Π_{LP}^k is optimal, we know that all the feasible paths in this network have a non-positive reduced cost. Consequently, to enumerate all the schedules with a null reduced cost, it suffices to solve the longest resource constrained path problem with the reduced costs as

arc costs and the following dominance rule: eliminate L_i^2 if and only if $\bar{Z}_i^1 > \bar{Z}_i^2$ and $T_i^1 \leq T_i^2$, where $L_i^1 = (\bar{Z}_i^1, T_i^1)$ and $L_i^2 = (\bar{Z}_i^2, T_i^2)$ are two labels corresponding to partial paths between $o(k)$ and node i, and the component $\bar{Z}_i$ of these labels specifies the reduced cost of the associated partial path.

The second approach consists of the following steps:

1. Keep the dual solution Π_{LP}^k of the linear relaxation if this one has an optimal value equal to S_k^*. Otherwise, obtain such a solution Π_{LP}^k by solving the linear relaxation without allowing the generation of schedules for employee k whose score is greater than S_k^*.

2. Solve, using the generalized algorithm of Desrochers and Soumis (1988), the longest resource constrained path problem on G_k^k with the arc reduced costs and the dominance rule presented above. A list of labels is then obtained at node $d(k)$

3. Among these labels, keep only those associated with a schedule of cost S_k^*. They represent all the feasible residual schedules for employee k with a score S_k^*

Since the dominance rule used in the second approach is tighter than the one used in the first approach, the second approach is more efficient when there is an important gap between the values S_k^* and $\bar{S}_k$. Thus, we retained the second approach for creating the set $\widehat{\Omega}_k^k$.

3.4   Updating $\widehat{\Omega}_i^k$ and Ω_i^k

Solving the problem $MIP(k \rightarrow k)$ provides an upper bound on the score of the best schedule for employee k. The computed optimal dual solution of its linear relaxation can also be used to eliminate some schedules from the sets $\widehat{\Omega}_i^k$, $i = 1, \ldots, k-1$ as suggested in the following proposition.

Proposition 3 *Assume that $MIP(k \rightarrow k)$ is feasible, its optimal value is S_k^* and that of its linear relaxation is S_k^{LP}. Let $\tilde{X}_{LP}$ and $\tilde{\Pi}_{LP}$ be a pair of primal and dual optimal solutions for its linear relaxation, and consider any variable x_{ij}, $i = 1, \ldots, k-1$, $j \in \widehat{\Omega}_i^k$, whose reduced cost $\bar{c}_{ij}$ with respect to $\tilde{\Pi}_{LP}$ is strictly less than $S_k^* - S_k^{LP}$. There exists no feasible solution to $MIP(k \rightarrow k)$ in which $x_{ij} = 1$ and whose value is S_k^*.*

Proof: Assume that there exists a feasible solution $\tilde{X}_{MIP}$ to $MIP(k \rightarrow k)$ in which $x_{ij} = 1$ and denote its value by $\tilde{Z}_{MIP}$. We will show that $\tilde{Z}_{MIP} < S_k^*$. First, notice that x_{ij} is a nonbasic variable in $\tilde{X}_{LP}$ because its reduced cost is not null. Therefore, as in the proof of Proposition 2, we can write

$$\tilde{Z}_{MIP} = S_k^{LP} + \bar{c}_H \tilde{X}_{MIP,H}, \tag{30}$$

where $\tilde{X}_{MIP,H}$ is the vector of the components of $\tilde{X}_{MIP}$ corresponding to the nonbasic variables in $\tilde{X}_{LP}$. Since $\bar{c}_H \leq 0$ ($\tilde{\Pi}_{LP}$ is optimal) and $\tilde{X}_{MIP,H} \geq 0$ ($\tilde{X}_{MIP}$ is feasible), we

obtain from (30) that

$$
\begin{aligned}
\tilde{Z}_{MIP} &\leq S_k^{LP} + \bar{c}_{ij} x_{ij} \\
&< S_k^{LP} + (S_k^* - S_k^{LP}) = S_k^*.
\end{aligned}
$$

∎

Consequently, to maintain a score of S_k^* for employee k, any schedule $j \in \widehat{\Omega}_i^k$ for any employee $i \in \{1, \ldots, k-1\}$ such that $\bar{c}_{ij} < S_k^* - S_k^{LP}$ cannot be assigned to the corresponding employee i. Thus, these schedule can be removed from the set $\widehat{\Omega}_i^k$.

By analyzing the schedules in the set $\widehat{\Omega}_\ell^k$, $\ell \in \{1, \ldots, k\}$, it might also be possible to reduce the sets $\widehat{\Omega}_i^k$, $i = 1, \ldots, k$, $i \neq \ell$, and the sets Ω_i^k, $i = k+1, \ldots, m$. Indeed, a pairing may be identified as essential to construct a best-score schedule for employee ℓ.

Definition
A pairing $p \in P$ is said to be *essential for employee* $\ell \leq k$ if $a_{\ell jp} = 1$, $\forall j \in \widehat{\Omega}_\ell^k$.

When a pairing p becomes essential for b_p employees, then it should not be included in the schedules of any other employees as stated in the following proposition.

Proposition 4 *Assume that $MIP(k \to k)$ is feasible and that its optimal value is S_k^*. Let $p \in P$ be an essential pairing for b_p employees and denote by E_p the set of these employees. There exists no feasible solution to $MIP(k \to k)$ whose value is S_k^* and for which $x_{ij} = 1$ where $i \in \{1, \ldots, m\} \setminus E_p$, $j \in \widehat{\Omega}_i^k$ or $j \in \Omega_i^k$ (depending on the value of i), and $a_{ijp} = 1$.*

The proof of Proposition 4 is omitted since it is trivial. This proposition allows updating the sets $\widehat{\Omega}_i^k$ and Ω_i^k by removing from these sets all schedules that satisfy the conditions stated above. Since column generation is used to generate the schedules in the sets Ω_i^k, the networks G_i^k must also be updated by removing all arcs representing a pairing p that is essential for b_p employees.

3.5 Integrality and infeasibility

When the set $\widehat{\Omega}_\ell^k$ for $\ell \in \{1, \ldots, k\}$ contains a single schedule j_ℓ, i.e., $| \widehat{\Omega}_\ell^k | = 1$, this schedule is the only residual schedule left for employee ℓ with a score S_ℓ^*. Given the loss of flexibility in the choice of a schedule for this employee, we propose to verify that a partially integer feasible solution exists when j_ℓ is chosen and the scores for the first k employees $(i = 1, \ldots, k)$ are set to their computed upper bound S_i^*. In fact, we propose to solve the following feasibility problem, $MIP(1 \to k)$, that involves integrality requirements on the variables associated with the schedules of the first k employees:

$$(MIP(1 \rightarrow k)) \begin{cases} \qquad Max\ Z = 0 & (31) \\ \text{subject to} \\ \displaystyle\sum_{i=1}^{k} \sum_{j \in \widehat{\Omega}_i^k} a_{ijp} x_{ij} \\ \displaystyle + \sum_{i=k+1}^{m} \sum_{j \in \Omega_i^k} a_{ijp} x_{ij} \ =\ b_p \qquad p = 1, \dots, P & (32) \\ \displaystyle\sum_{j \in \widehat{\Omega}_i^k} x_{ij} \ =\ 1 \qquad i = 1, \dots, k & (33) \\ \displaystyle\sum_{j \in \Omega_i^k} x_{ij} \ =\ 1 \qquad i = k+1, \dots, m & (34) \\ x_{ij} \ \in\ \{0,1\} \quad i = 1, \dots, k, \\ \qquad\qquad\qquad \forall j \in \widehat{\Omega}_i^k & (35) \\ x_{ij} \ \geq\ 0 \qquad i = k+1, \dots, m, \\ \qquad\qquad\qquad \forall j \in \Omega_i^k & (36) \end{cases}$$

Problem $MIP(1 \rightarrow k)$ is also solved by the branch-and-price algorithm used to solve $MIP(k \rightarrow k)$. On the one hand, when this problem is feasible, we know that schedule j_ℓ is compatible with best-score schedules for the other first $k-1$ employees $i = 1, \dots, k$, $i \neq \ell$. In this case, the solution process moves on to the next employee after fixing the schedule j_ℓ for employee ℓ as in the solution approach of Gamache et al. (1998b). Note that several schedules can be fixed at the same time if $\mid \widehat{\Omega}_\ell^k \mid = 1$ for several values of $\ell \in \{1, \dots, k\}$. On the other hand, when $MIP(1 \rightarrow k)$ is infeasible, this indicates that at least one of the upper bounds S_i^*, $i = 1, \dots, k$ overestimates the optimal score of the corresponding employee, as shown by the following proposition.

Proposition 5 *Assume that problem $IP(1)$ is feasible (that is, all problems $IP(k)$, $k = 1, \dots, m$, are feasible according to Proposition 1) and denote the optimal value of $IP(k)$ by Z_{IP}^k. If $MIP(1 \rightarrow k)$ is infeasible, then $Z_{IP}^\ell < S_\ell^*$ for at least one employee $\ell \in \{1, \dots, k\}$.*

Proof: The proof proceeds by contradiction. Assume that, for all employees $\ell = 1, \dots, k$, $Z_{IP}^\ell = S_\ell^*$. Then it easy to see that, in this case, an optimal solution for $IP(k)$ is also a feasible solution for $MIP(1 \rightarrow k)$. This contradicts the infeasibility of $MIP(1 \rightarrow k)$. ∎

Therefore, when $MIP(1 \rightarrow k)$ is infeasible, one must backtrack to find the overestimating upper bound. We noticed that this upper bound is often the current one S_k^*. To verify that this is the case, one can solve the problem $IP(k)$. However, this might be very time consuming. Instead we resort to solving a modified $MIP(k \rightarrow k)$ problem that forces the optimal score of the employee k to be smaller than or equal to $S_k^* - 1$. This is done by adding a cut in the subproblem of employee k (see Gamache *et al.*, 1998b). The solution process then continues as if $MIP(k \rightarrow k)$ was solved for the first time. In practice, very few modified $MIP(k \rightarrow k)$ problems are solved for the same employee.

At any time during the solution process, a problem $MIP(k \to k)$ might be found to be infeasible. The next proposition points out the cause of this infeasibility.

Proposition 6 *Assume that problem $IP(1)$ is feasible (that is, all problems $IP(k)$, $k = 1, \ldots, m$, are feasible according to Proposition 1) and denote the optimal value of $IP(k)$ by Z_{IP}^k. If $MIP(1 \to k)$ is infeasible, then $Z_{IP}^\ell < S_\ell^*$ for at least one employee $\ell \in \{1, \ldots, k-1\}$.*

Proof: The proof proceeds by contradiction. Assume that, for all employees $\ell = 1, \ldots, k-1$, $Z_{IP}^\ell = S_\ell^*$. Then it easy to see that, in this case, an optimal solution for $IP(k)$ is also a feasible solution for $MIP(k \to k)$ since this problem is a relaxation of $IP(k)$. This contradicts the infeasibility of $MIP(k \to k)$.   ■

When $MIP(k \to k)$ is infeasible, one could try to solve a modified $MIP(k-1 \to k-1)$ problem with an additional cut to see if S_{k-1}^* is the wrong upper bound. However, since a $MIP(k \to k)$ problem is less constrained than a $MIP(1 \to k)$ problem, it is less obvious that the wrong upper bound is S_{k-1}^*. Consequently, very often, a long series of modified $MIP(k-1 \to k-1)$ problems would need to be solved to finally yield an infeasible $MIP(k-1 \to k-1)$ and move on to the preceding employee $k-2$. To avoid this long series of problems, the proposed approach backtracks by solving the previous $IP(i)$ problems, for $i = k-1, k-2, \ldots, \ell$, until finding one $(IP(\ell))$ that is feasible if one exists. This can also be very time consuming, but it seems preferable. If no feasible $IP(\ell)$ problem exists, the overall problem is infeasible.

The problems $IP(i)$ are also solved by branch-and-price. Note that, once a feasible $IP(\ell)$ problem is found, the set $\widehat{\Omega}_\ell^\ell$ can be created and the sets $\widehat{\Omega}_i^\ell$, for $i = 1 \ldots, \ell - 1$, and Ω_i^ℓ, for $i = \ell + 1, \ldots, m$, can be updated from the solution of $IP(\ell)$, as explained in Sections 3.3 and 3.4. Finally, note that, when the computed solution for $MIP(m \to m)$ is fractional, the problem $MIP(m \to m)$ needs to be solved to obtain an integer solution.

4 Computational Results

The exact approach described in the previous section was tested on real-life cases corresponding to the December 2001 PBS problems for the pilots at Air Canada. Table 1 reports the results obtained for several fleets (Airbus A320, A340, Boeing 767 and CRJ) at several bases (Montreal (yul), Toronto (yyz), Vancouver (yvr), and Winnipeg (ywg)). These problems have been chosen for their respective size and also because December is known in the airline industry for being one of the most difficult months for constructing monthly work schedules since employee preferences are often similar. Table 1 presents, for each problem, an overview of the results. It shows the improvements of the solutions by comparing the scores of th schedules constructed using the new approach with those obtained by the method developed by Gamache et al. (1998b). In this table, the second column indicates the average number of columns with a score equal to S_k^* that was enumerated at each iteration. Th third one, entitled *the number of unassigned employees*, indicates the average number of employees whose se $\widehat{\Omega}_i^k$, $i \leq k$, contains more than one schedule at each

iteration. The fourth one gives the average number of essential tasks identified at each iteration. The fifth column shows the relative increase of the solution value for the first employee (denoted α) whose schedule's score has been modified compared to the previous method. The first improvement is calculated as follows:

$$\text{First Improvement} = \frac{S_\alpha^{*\ New} - S_\alpha^{*\ Old}}{S_\alpha^{*\ Old}}$$

where $S_\alpha^{*\ New}$ and $S_\alpha^{*\ Old}$ are the score of the best schedule assigned to α in the new and the old (Gamache *et al.*, 1998b) approaches, respectively. Sinc scores are based on a variable scale which differs from one pilot to another, this measurement is not always the best one for showing the amplitude of the improvement, especially if $S_\alpha^{*\ Old}$ is a huge number. In order to have a better indicator of the improvement, a second ratio is used. In this ratio, S_α^{Best} indicates the score of the best schedule that employee α can have at iteration α. This score is evaluated by solving the longest path problem in the network associated with employee α while taking into account the schedules already assigned to the most senior employees at the previous iterations. According to this process, S_α^{Best} is an upper bound on the score of the schedule of that employee and constitutes a stopping criteria during the solution of th $MIP_{\alpha \to \alpha}(\alpha)$. The sixth column, called the improvement on the upper bound, indicates the relative increase of the value of the solutions by calculating the following ratio

$$\text{Bound Improvement} = \frac{S_\alpha^{*\ New} - S_\alpha^{*\ Old}}{S_\alpha^{Best} - S_\alpha^{*\ Old}}.$$

This ratio indicates, in percentage, the improvement of the solution value compared with the best possible improvement.

One should note that from employee $\alpha + 1$ to employee m, the old and the new approaches are solving different problems; thus, any comparison between the two approaches is worthless. However, in order to have an indicator for the overall improvement of the new approach, in the last two columns of the table we indicate the number of employees whose schedule has been improved by this approach and those whose schedule has been deteriorated.

One notices that there is a net improvement in the score of the schedules assigned to employees when using this new approach. For example, the first improvement is positive in all cases and increases to 25% in the best case. However, an improvement in the solution obtained for more senior employees inevitably involves a degradation of the scores of some more junior employees whose preferences are quite similar. This result is in accordance with the lexicographic objective of this problem where employees having a higher seniority are advantaged. However, it can be mentioned that the ratio of the number of schedules that have been improved over those that have been deteriorated is, on average, always greater than 2.33.

Besides, the most significant improvement of the new approach over the old one is observed when one compares how far the value of the solution obtained by the new approach

Table 1: Improvement of solution

Problem	Solution process			Solution quality			
(Nb of employees)	Nb of columns enumerated	Nb of unassigned employees	Nb of essential tasks	First improvement (%)	Bound improvement (%)	Nb of improved scores	Nb of deteriorated scores
yyz-340(91)	5.59	5.78	3.98	14.4	52.67	31	10
yul-crj(36)	8.16	1.09	10.96	1.46	20.11	16	2
yyz-crj(91)	9.56	12.34	9.92	1.55	57.5	30	5
yul-320(90)	10.83	5.14	4.66	5.06	48.78	28	9
yvr-737(91)	40.34	5.93	4.21	5.21	68.47	22	9
yvr-320(68)	76.81	5.16	3.54	3.77	60.11	18	7
ywg-320(56)	4.2	2.7	4.2	0.01	70.55	23	5
ywg-crj(17)	2	1.33	3	24.99	65	4	0

is from the value of the best possible solution, i.e. the upper bound. When using the new approach, the average improvement is greater than 20% and it reaches 70% in some cases.

It is also important to note that the average number of employees to whom a schedule has not yet been assigned remains small; i.e. on average this number is 4.9. This can be explained by the identification of an important number of essential tasks at each iteration. Since the number of optimal schedules enumerated at each iteration and the size of problem $LP(k)$ solved at each iteration are small, the solution time does not increase considerably when using this optimal method compared to the old approach.

Table 2 compares the solution times (in seconds) of both approaches (Old and New). The table first presents the solution time (in seconds) for solving $MIP(k \to k)$ and $MIP(1 \to k)$. The next two columns show the total time spent solving $IP(k)$ when backtracks were necessary and also the number of times they were needed by both approaches. The seventh column indicates the time spent enumerating columns of $\hat{\Omega}_k^k$. Finally, the last two columns indicate the total solution time.

This comparison can be divided into two groups: those cases associated with the problems that are easy to solve (yyz-340, yul-crj, yvr-737, yvr-320, and ywg-crj) and those associated with the more difficult ones (yyz-crj, yul-320, and ywg-320). In the first group, one notices that the solution time increases with the new approach and this increase is due essentially to the time devoted to the solution of $MIP(1 \to k)$. However, the solution time remains quite small in spite of the increase. However, for the second group of problems, i.e. those that needed the backtracking process in Gamache *et al* (1998b), the new optimal approach is competitive with the old one and moreover the number of backtracks is reduced. This results in a significative reduction of the solution time. Indeed, the backtrack is due essentially to a bad decision taken during the solution of $MIP(k \to k)$. Since no schedules

Table 2: Solution times in seconds

Problem	$MIP(k \to k)$		$MIP(1 \to k)$	Backtrack		Enum	Total Time	
(Nb of employees)	Old	New	New	Old (Nb)	New (Nb)	New	Old	New
yyz-340(91)	147	170	102	-	-	163	210	492
yul-crj(36)	148	1302	18	-	86(2)	381	187.17	2307
yyz-crj(91)	114115	415878	1955	545034(14)	36508(1)	2768	659935	457780
yul-320(90)	1875	26953	3394	18286(9)	36203(8)	1863	20448	69340
yvr-737(91)	117	189	30	-	-	144	182	481
yvr-320(68)	68	79	8	-	-	44	80	126
ywg-320(56)	145	147	7	1354(7)	-	78	1476	189
ywg-crj(17)	0.71	0.83	0.01	-	-	0.31	0.73	1.08

are chosen *a priori* then the solution approach has more flexibility for the construction of schedules for the most junior employees. In general, the number of backtracks has been reduced when using the new method (excepted for problem yul-crj-ca) and in some cases backtracks have been eliminated. Problem ywg-320-ca is a good example where the new approach performs quite well avoiding the backtracks while 7 backtracks that were needed with the old approach.

5   Conclusion

We have presented an exact sequential approach for solving the PBS problem in the airline industry. The implementation of this new approach has considerably improved the schedule of the pilots at Air Canada in comparison to the current approach which, to our knowledge, is the best currently available. In fact, better schedules are constructed for several employees. Although, the solution time has increased in some cases, it remains acceptable or is even less for some of the most difficult problems. This is due to the fact that the new approach often reduces the number of backtracks.

References

Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance (1998). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 46, 316–329.

Byrne, J. (1988). A Preferential Bidding System for Technical Aircrew. *1988 AGIFORS Symposium Proceedings* 28, 87–99.

Desaulniers, G., J. Desrosiers, I. Ioachim, F. Soumis, M.M. Solomon, and D. Villeneuve (1998). A Unified Framework for Time Constrained Vehicle Routing and Crew

Scheduling Problems. In: *Fleet Management and Logistics*, T.G. Crainic and G. Laporte (eds.), Kluwer, Norwell, MA, 57–93.

Desrochers, M. and F. Soumis (1988). A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows. *INFOR* 26, 191–212.

Gamache, M. and F. Soumis (1998a). A Method for Optimally Solving the Rostering Problem. In: *Operations Research in the Airline Industry*, G. Yu (ed.), Kluwer, Norwell, MA, 124–157.

Gamache, M., F. Soumis, J. Desrosiers, D. Villeneuve, and E. Gélinas (1998b). The Preferential Bidding System at Air Canada. *Transportation Science* 32, 246–255.

Gamache, M., F. Soumis, G. Marquis, and J. Desrosiers (1999). A Column Generation Approach for Large-Scale Aircrew Rostering Problems. *Operations Research* 47, 247–262.

Giafferri, C., J.P. Hamon, and J.G. Lengline (1982). Automatic Monthly Assignment of Medium-Haul Cabin Crew. *1982 AGIFORS Symposium Proceedings* 22, 69–95.

Glanert, W. (1984). A Timetable Approach to the Assignment of Pilots to Rotations. *1984 AGIFORS Symposium Proceedings* 24, 369–391.

Gontier, T. (1985). Longhaul Cabin Crew Assignment. *1985 AGIFORS Symposium Proceedings* 25, 44–66.

Irnich, S. and G. Desaulniers (2004). Shortest Path Problems with Resource Constraints. Technical report, *Les Cahiers du GERAD* G-2004-11, HEC Montréal, Canada.

Marchettini, F. (1980). Automatic Monthly Cabin Crew Rostering Procedure. *1980 AGIFORS Symposium Proceedings* 20, 23–59.

Mayer, M. (1980). Monthly Computerized Crew Assignment. *1980 AGIFORS Symposium Proceedings* 20, 93–124.

Moore, R., J. Evans, and H. Ngo (1978). Computerized Tailored Blocking. *1978 AGIFORS Symposium Proceedings* 18, 343–361.

Nicoletti, B. (1975). Automatic Crew Rostering. *Transportation Science* 9, 33–42.

Ryan, D.M. and J.C. Falkner (1988). On the Integer Properties of Scheduling Set Partitioning Models. *European Journal of the Operational Research* 35, 442–456.

Ryan, D.M. (1992). The Solution of Massive Generalized Set Partitioning Problems in Air Crew Rostering. *Operations Research* 45, 649–661.

Ryan, D.M. and R.D. Day (1997). Flight Attendant Rostering for Short-Haul Airline Operations. *Journal of the Operational Research Society* 43, 459–467.

Sarra, D. (1988). The Automatic Assignment Model. *1988 AGIFORS Symposium Proceedings* 28, 23–37.

Sherali, H.D. and A.L. Soyster (1983). Preemptive and Non preemptive Multi-Objective Programming: Relationships and Counterexamples. *Journal Of Optimisation Theory and Applications* 39, 173–186.

Tingley, G.A. (1979). Still Another Solution Method for the Monthley Aircrew Assignment Problem. *1979 AGIFORS Symposium Proceedings* 19, 143–203.