Les Cahiers du GERAD ISSN: 0711–2440

Branch and Cut at the Subproblem Level in a Column Generation Approach: Application to the Airline Industry

H. Achour, M. Gamache
F. Soumis

G–2003–34

May 2003

Branch and Cut at the Subproblem Level in a Column Generation Approach: Application to the Airline Industry

Heykel Achour **Michel Gamache**

François Soumis

GERAD and
École Polytechnique de Montréal
C.P. 6079 succ. Centre-ville
Montréal (Québec) Canada
H3C 3A7

May, 2003

Les Cahiers du GERAD

G–2003–34

Abstract

This paper presents a new branching strategy that is applied on the cost of a subproblem during the solution of a large-scale linear program by a column generation technique. This branch and cut strategy has been used to improve the solution time for the preferential bidding problems encountered in the airline industry. Moreover, it is shown that this strategy can also be applied to other problems with particular structures.

Résumé

Nous proposons une stratégie de type branchement et coupe intervenant au niveau des sous-problèmes dans le cadre de la procédure de génération de colonnes. L'application de cette stratégie pour la résolution des problèmes de fabrication des horaires personnalisés avec priorités chez Air Canada présentant un grand gap d'intégrité a permis d'améliorer le temps de calcul et de générer de meilleurs horaires pour un bon nombre de pilotes. La taille et la profondeur de l'arbre de branchement ont été réduites de façon très significative. Nous avons discuté de l'extension de la stratégie pour la résolution d'autres classes de problèmes. Plusieurs applications connues dans la littérature sont présentées.

1 Introduction

The column generation technique has been commonly used to solve large scale optimization problems. This decomposition technique consists of solving alternatively a master problem and one or more subproblems. We consider the case where subproblems generate feasible paths, called path variables, which may be integrated into the master problem during the next iteration. This solution approach considers implicitly all the variables of the problem but generates only a small subset of them. The main role of the master problem consists of finding the optimal solution of the current restricted problem, i.e. the problem composed of the variables that have been generated, and also of providing the dual variables associated with this current optimal solution. Subproblems are used on the one hand to verify if the current solution of the master problem is optimal and on the other hand to generate variables that will be candidates for entering into the basis of the master problem when the solution is not optimal. These subproblems are formulated as shortest path problems on an acyclic network. This procedure is imbedded in a branch-and-bound algorithm when integrality constraints are involved on the path variables. At each node of the branch and cut strategy, the column generation technique is used to find the optimal solution of the linear relaxation of the problem. The value of this solution is used as a lower or an upper bound depending on if the master problem is a minimization one or a maximization one, respectively. In several applications, there are also constraints ensuring that the path variables of each subproblem form a convex combination. In those cases, a generic relaxed linear program can be formulated as follow:

$$\min(\max)\ \mathbf{c}^T\mathbf{x} \tag{1}$$

subject to

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{2}$$

$$\sum_{j \in SP_k} x_j = 1 \ \forall k \tag{3}$$

$$\mathbf{x} \in [0,1]. \tag{4}$$

where SP_k represents the k^{th} subproblem.

The particularity of this formulation is that the value of the solution of the restricted master problem can be used to accelerate the solution of the subproblems. Sometimes, it is mandatory to use this value as a bound in the subproblem to obtain an optimal integer solution. This is the case in the monthly work schedule construction problem for pilots in the airline industry when preferences are taken into account and priority order between employees must be respected (see Gamache *et al*, 1998b). This problem, known as the PBS (preferential bidding system) problem in the airline industry, will be used to present a new branch and cut strategy that can be applied to any problem having the same structure as the generic linear program described by equations (1) to (4). Sections 2 and 3 give a description of the PBS problem and a summary of the solution approach proposed by Gamache *et al* (1998b), respectively. The new branch and cut strategy is detailed in Section 4. Section 5 presents a comparison of the efficiency of the new branch and cut strategy versus the previous approach used by Gamache *et al* (1998b). Finally, generalization of this method to other applications is presented in Section 6.

2 A first application: The PBS problem in the airline industry

In the airline industry, the construction of monthly work schedules for crew members is a difficult problem to address because it involves large-scale problems: thousands of constraints and millions of variables. Moreover, if one has to consider a set of preferences expressed by each employee during the schedule construction, the problem becomes more complicated because the problem consists not only in covering a set of activities but also in satisfying employees as much as possible. The satisfaction of each employee is measured by assigning weights to activities in accordance with each employee preferences. For each schedule, the sum of all weights represents the *score* of the schedule.

The construction of monthly work schedules while taking into account the preferences of each crew member is generally done according to two modes. The first mode, called the *rostering problem*, consists in constructing schedules that maximize the total satisfaction of all employees. This type of schedule construction is used by Air France (Giafferri *et al.* 1982, Gontier 1985, Gamache *et al.* 1998a), Alitalia (Nicoletti 1975, Marchettini 1980, Sarra 1988, Federici et Paschina 1990), Lufthansa (Glanert 1984), SwissAir (Tingley 1979), Air New-Zealand (Ryan 1992) and El-Al Israel Airline (Mayer 1980).

The second mode of constructing schedules is called the *preferential bidding problem* and it is mostly used by North-American airlines. The preferential bidding problem is similar to the rostering problem except that the preferences must be assigned to employees according to a priority order. In this problem, the objective function consists of maximizing the satisfaction of the employees while giving priority to the most senior of them.

In both modes, during the construction of the schedules, planners must respect several constraints such as those included in the collective agreement between the company and the employees and those related to the airline security rules. They must also take into account activities that have been pre-assigned to employees, such as vacations and training periods. Moreover, the set of feasible schedules must cover all the *pairings* planned during the month. A pairing is a sequence of flights that starts from a city, called a base, and ends at this same city.

In the preferential bidding problem, schedules are constructed in a sequential manner according to the seniority order; i.e. at each iteration of the solution process, the schedule of only one employee is constructed. The use of a sequential approach for solving this problem is essential due to the large number of possible bids for each employee, the necessity of reflecting the seniority order between employees, and the large number of employees to consider. In fact, the differentiation of the preferences of all employees in a unique objective function would have necessitated numbers so huge that they could not have been used in a computer.

At each iteration of the sequential approach, the objective function changes to reflect the preferences of the employee whose schedule is constructed. Moreover, the schedule currently constructed must be compatible with those already constructed and must permit the assignment of the *residual pairings*, i.e. those that are not assigned to an employee yet, to the set of *residual employees*, i.e. the most junior employees whose schedules have not been constructed yet. This approach is used in order to make sure that the assignment of activities to one employee is not done to the detriment of more senior employees.

The preferential bidding problem is quite new and thus less known than the rostering problem. Therefore, there are only few papers in the literature that deal with this problem. Companies such as Quantas (Moore *et al.*, 1978), CP Air (Byrne, 1988), Midwest Express Airlines, Inc. (Schmidt, 1994), Air Canada (Gamache *et al.*, 1998) have developed their method to solve the preferential bidding problem. Because of the complexity of the problem, most of the methods presented in the literature are based on a greedy heuristic.

However, in 1998, Gamache *et al.* proposed an algorithm having the propriety of being optimal for each employee considering that schedules have been already assigned to the most senior employees. Like all the other approaches, this method uses a sequential process based on seniority. The next section will detail this method.

3 The solution approach for the PBS Problem

3.1 General description of the method

The method proposed by Gamache *et al.* (1998a) constructs the schedule of employees one after the other, starting from the most senior employee to the most junior one. At iteration k of the solution process, the schedules of the $k-1$ most senior employees have been already constructed. The method solves the residual problem that consists of an integer linear program which assigns the set of residual pairings to the set of residual employees. The objective function consists in maximizing the score of employee k's schedule.

Given m employees and a set of p pairings, the problem of constructing a monthly work schedule for one employee at each iteration of the sequential approach can be formulated as an integer linear program. At the first iteration of the solution process, the integer linear program includes a set of m constraints requiring one feasible schedule for each employee and a set of p constraints for the covering of each pairing.

At iteration k, the problem (IP_k) can be written as follows:

$$\max \sum_{j \in S_k^k} c_j x_j \tag{5}$$

subject to

$$\sum_{j \in S^k} a_{ij} x_j = b_i^k, \quad i = 1, \ldots, p \tag{6}$$

$$\sum_{j \in S_e^k} x_j = 1, \quad e = k, \ldots, m \tag{7}$$

$$x_j \in \{0, 1\}, \quad \forall j \in S^k \tag{8}$$

where

$$S^k = \bigcup_{e=k}^{m} S_e^k \text{ and}$$

- S_e^k is the set of all feasible schedules for employee e at iteration k;
- c_j is the score of schedule j;
- x_j is a binary variable which equals 1 if schedule j is chosen and 0 otherwise;
- a_{ij} equals 1 if pairing i is part of schedule j, 0 otherwise;
- b_i^k represents the number of employees still required by pairing i at iteration k.

The optimal solution to this problem gives the best schedule for the employee k while also ensuring a feasible solution for all the residual employees.

This problem is solved by a column generation technique imbedded in a branch-and-bound algorithm. The master problem solves the linear relaxation of problem IP_k (equations (5) to (7)), called LP_k, on a subset of variables (feasible schedules). Subproblems are used to produce feasible schedules. A subproblem is associated with each of the residual employees from k to m. Each subproblem consists of solving a shortest path problem (longest path problem) with resource constraints on a graph that takes into account the particularities of each employee.

Given $G_e^k = G(N_e^k, A_e^k)$ the residual graph of employee e at iteration k, $\mathcal{X}_{ij}(e)$ the flow variables on A_e^k and $T_i^r(e), r \in R$, the resource variables on N_e^k, the mathematical formulation of the subproblem is then defined as follows:

$$\max \sum_{(i,j)\in \mathcal{A}_e^k} \bar{c}_{ij}(e)\mathcal{X}_{ij}(e) \tag{9}$$

subject to

$$\sum_{j\in N_e^k} \mathcal{X}_{ij}(e) - \sum_{j\in N_e^k} \mathcal{X}_{ji}(e) = \begin{cases} +1, & i = source \\ 0, & \forall i \in N_e^k \\ -1, & i = sink \end{cases} \tag{10}$$

$$\mathcal{X}_{ij}(e)\left(T_i^r(e) + t_{ij}^r - T_j^r(e)\right) \leq 0, \quad \forall (i,j) \in A_e^k, \forall r \in R \tag{11}$$

$$\ell_i^r \leq T_i^r(e) \leq u_i^r, \quad \forall i \in N_e^k, \forall r \in R \tag{12}$$

$$\mathcal{X}_{ij}(e) \geq 0, \quad \forall (i,j) \in A_e^k \tag{13}$$

$$\mathcal{X}_{ij} \in \{0,1\}, \quad \forall (i,j) \in A_e^k \tag{14}$$

In this model,

$$\bar{c}_{ij}(e) = \begin{cases} w_{ij}(e) - \pi_{ij}(e), & \text{for } e = k \\ -\pi_{ij}(e), & \text{for } e = k+1,\ldots,m \end{cases} \tag{15}$$

where $w_{ij}(e)$ is the weight attributed to the task represented by arc (i,j) of subproblem e and $\pi_{ij}(e)$ represents the sum of the dual variables that apply on arc (i,j).

The objective function (9) of subproblem k maximizes the marginal score of employee k's schedule. Constraints (10) and (13)-(14) are associated with the flow on the network. The inequalities (12) evaluate at each node i of the graph if the amount of each resource accumulated along the path between the source node and node i, i.e. $T_i^r(e)$, falls between a lower bound l_i^r and an upper bound u_i^r. The resource constraints are used for cumulating the number of flight credits, the number of consecutive working days, the duration of rest periods, the fatigue index, etc. Finally, the set of inequalities (11) ensures the compatibility between the flow and the resources.

At each iteration of the proposed optimization method, an integer program IP_k is solved. Since the objective function only considers the preferences of employee k, the solution of IP_k gives the maximum score schedule to the k^{th} employee and assigns a feasible schedule for all employees from $k+1$ to m. In fact, the integer solution found at each iteration has for its only goal to ensure that the assignment of the best schedule to employee k will lead to an integer feasible solution.

The solution to each IP_k may prove to be very long; when considering that this problem has to be solved m times, where m is the number of employees, e.g. 300 to 3000 in some instances. Moreover, in the optimal solution of IP_k, the schedules constructed for employees $k+1$ to m are useless for the subsequent iterations because of the perpetual change of the objective function at each iteration. Thus, a new strategy has to be considered.

To reduce the solution time, the following approach was proposed by Gamache *et al.* (1998). At each iteration, a MIP_k is solved instead of an IP_k. From the optimal solution of the linear relaxation of the generalized set partitioning problem, a branch-and-bound algorithm is used to find an integer solution only for the variables associated with employee k. The choice of this strategy is based on the hypothesis that most of the time the best schedule of employee k obtained from the optimal solution of the MIP_k problem is also part of the optimal solution of the IP_k. Of course, this situation will not always be true and an infeasible solution to a mixed integer program (MIP_k) will sometime be found. In such a situation, one must backtrack and find an optimal solution to IP_{k-1}, if possible; otherwise, this backtracking process is repeated until a feasible solution is obtained for IP_ℓ where $\ell < k$.

3.2 The use of cuts at the subproblem level

Even when using this strategy, the solution of MIP_k using a branch-and-bound algorithm may be time-consuming because the integrality gap is sometime very large (almost 100% in some cases). This is usually the case when the solution to LP_k is fractional and occurs as a convex combination of schedules with different scores; i.e. when $Z^k_{LP} \in [Z_{INF}, Z_{SUP}]$ where Z_{INF} and Z_{SUP} are the lowest and the highest scores among the schedules corresponding to non zero basic variables (see Figure 1). In order to reduce the solution time in such situations, Gamache *et al.* (1998) have proposed to use cuts to improve the upper bound on the score of employee k, i.e. Z_{SUP}. If Z^k_{LP} is the optimal value of the solution of LP_k, then Z^k_{LP} constitutes an upper bound on the maximum score of employee k's schedules for which IP_k is feasible. The maximum score Z^k_{LP} is expressed as a convex combination of several schedules having either similar or different scores. If these scores are different, i.e. $Z_{INF} \neq Z_{SUP}$, then some of them are strictly greater than the value of Z^k_{LP} (see Figure 1); consequently, these schedules cannot be part of any integer solution of IP_k and thus can be removed from the domain of the feasible schedules of employee k without deteriorating the optimal solution.

This deep cut eliminates not only the fractional solution, but some integer solutions that are generated by the subproblem but that will never be part of a feasible solution of the master problem. It eliminates all the schedules whose score is greater than Z^k_{LP} and reduces consequently the domain of feasible schedules. These cuts are introduced as soon as needed until the solution of the linear relaxation becomes a convex combination of schedules having an identical score, i.e. $Z_{INF} = Z_{SUP}$. Note that the cut applies only to the schedules of employee k, so that it is applied locally during the solution of subproblem k. To implement the cut, an additional resource indicating the score of the path is introduced during the solution of the longest path problem with resource constraints in order to forbid the generation of columns (schedules) having a score greater than the value of the linear relaxation. This additional resource is written as follows:

$$\mathcal{X}_{ij}(k)\,(W_i(k) + w_{ij} - W_j(k)) \leq 0 \qquad \forall (i,j) \in A^k_k \tag{16}$$

$$W_i(k) \leq \lfloor Z^k_{LP} \rfloor \quad \forall i \in N^k_k. \tag{17}$$

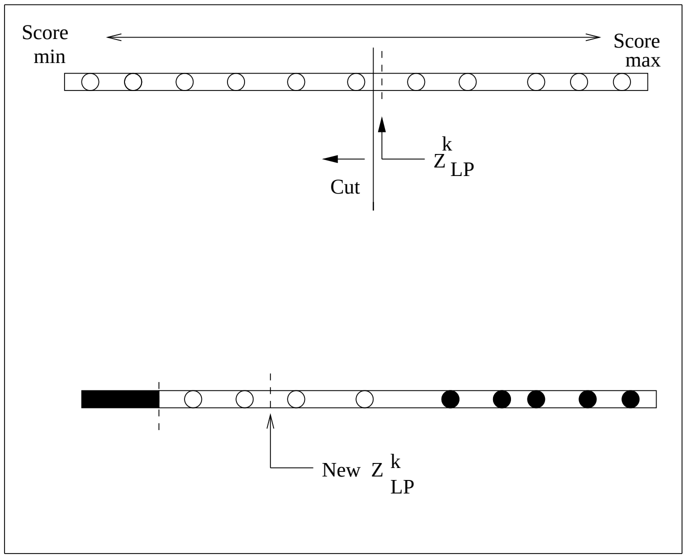

Figure 1: Illustration of the cut and the basis of the optimal solution of LP_k

At each node of subproblem k's graph, the current score of the path, $W_i(k)$, is compared with an upper bound on the resource which is equal to the current value of the linear relaxation (17). The addition of this resource to the subproblems allows the generation of feasible schedules during the solution of the preferential bidding problem.

Without the use of this cut for some employees, it was impossible to reduce the integrality gap. In one case, this approach permitted eliminating an integrality gap of 90% with only 7 cuts. To the knowledge of the authors of this paper, it was the first time such a cut was proposed in a subproblem of a column generation approach.

4 A new strategy: A branching decision

4.1 Some drawbacks when using cuts

The introduction of a cut in subproblem k necessitates, as explained in Section 3, the addition of a new resource on the score of the generated paths. However, the addition of this constraint increases considerably the solution time of the longest path problem with resource constraints. Indeed, the number of non dominated paths (labels) becomes large when using the cut since the new resource is positively correlated with some resources and particularly with the reduced cost. The number of labels generated in subproblem k increases considerably, which increases the solution time of subproblem k.

To simplify, let us consider the solution of a longest path with only one resource: the score. Let us consider two sub-paths of the network of employee k incoming at node i. Also, let $(\overline{C}_i^1, W_i^1)$ and $(\overline{C}_i^2, W_i^2)$ be the labels associated with these two sub-paths where $\overline{C}_i^j$ represents the value of the reduced cost of the sub-path j at node i and W_i^j the value of its score at node i. The sub-path 1 is dominated by sub-path 2 and can be eliminated from the sub-problem if $\overline{C}_i^2 \geq \overline{C}_i^1$ and $W_i^2 \leq W_i^1$. It is clear that if both elements of the vector are positively correlated, then the reduced cost and the score change in the same direction and the set of labels at each node forms a slim cluster of points having a positive slope (see Figure 2). There are several non dominated labels along the upper envelope (border) because at each node of subproblem k, one tries to maximize the reduced cost while also maintaining the score below a certain value. Based upon this observation, the addition of the cut on the score reduces the elimination by dominance of sub-paths during the solution of the longest path problem.

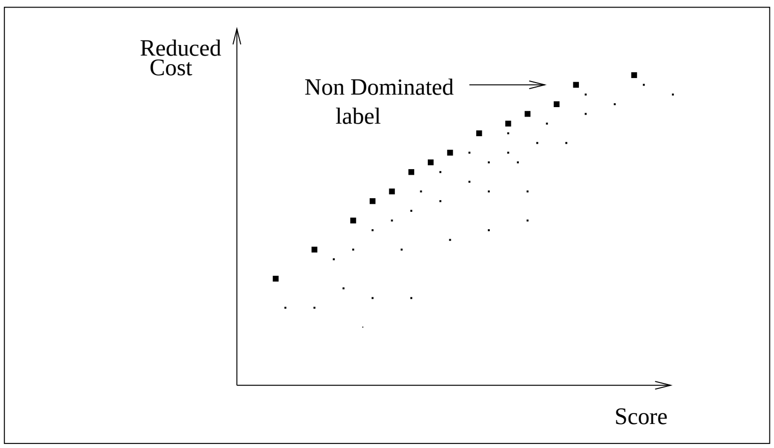

Figure 2: Dominance when using the cut

4.2 Description of the new approach

The objective of this paper is to propose a new solution strategy having the same effectiveness for reducing the integrality gap as the use of the cuts in the subproblem but without the drawbacks on the solution time.

When using the cuts, the main idea was to reduce the interval $[Z_{INF}, Z_{SUP}]$ by using a constraint on the score of the generated columns. Reducing this interval by a half at each iteration ensures a rapid convergence. The innovation of the new approach is to use a lower bound on the score instead of an upper bound to reduce this interval. As it will be shown in the following sections, this new approach considerably reduces the solution time while also keeping the same number of iterations.

The new solution approach combines branching decisions and cuts. Branching decisions remove integer solutions from the domain of feasible solutions of the master problem (see Figure 3) which was not the case when using cuts. Figure 3 presents a schematization in 2 dimensions of the domain of feasible solutions delimited by the constraints of the master problem and those of subproblem k. In this figure, regions (1) and (3) show the parts of the domain that are removed by the branching decision and the cut, respectively.

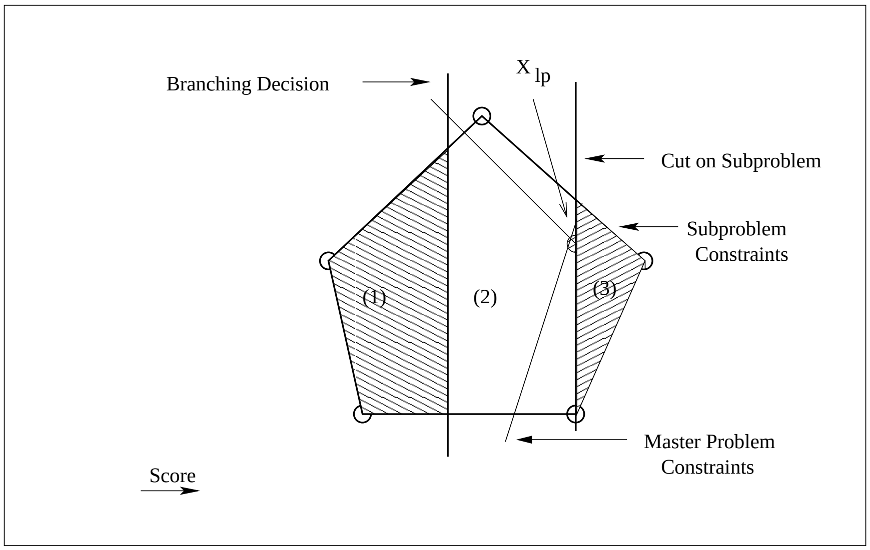

Figure 3: Illustration of the new branching decision and the domain of feasible solutions

This branching strategy consists in imposing in subproblem k the generation of columns whose score is greater than or equal to a given value denoted K_ℓ, where K_ℓ represents the minimum score of the schedules to be generated by subproblem k when the ℓ^{th} branching decision is taken. As in the addition of the cut, the branching decision will be implemented directly into the solution of the longest path problem by using a new resource. Since the implementation of the longest path algorithm with resource constraints does not take into account the lower bound on the resource, the new resource has been implemented as follows:

$$-W_i(k) \quad \leq \quad -K_\ell \quad \forall i \in N_k^k \tag{18}$$

This new resource is now negatively correlated with some of the resources and principally with the reduced cost. Therefore, it permits this time a more aggressive elimination of labels by dominance in the subproblem (see Figure 4). Thus, the imposition of this resource reduces considerably the number of sub-paths generated by the subproblem. This will permit improving the solution time of the longest path problem with resource constraints.

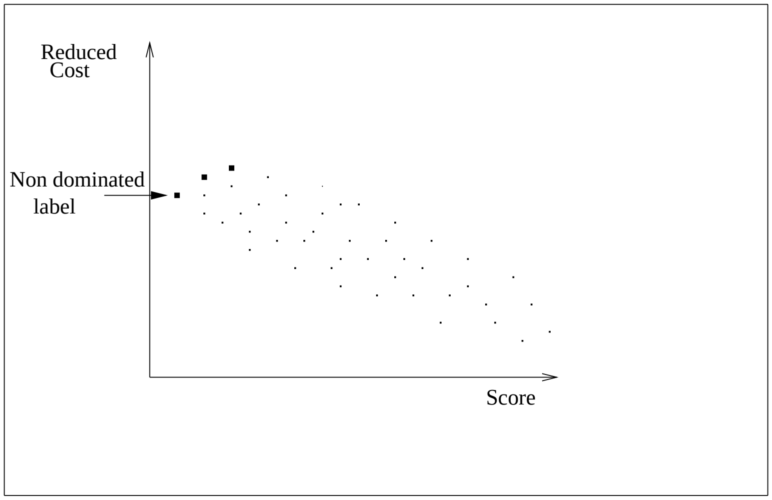

Figure 4: Dominance with the new strategy

4.3 Some propositions

In order to detail the application of this new branching strategy, we first present three propositions.

Proposition 1

Let (LP_k^1) be the linear relaxation problem obtained from (LP_k^0) after adding a first branching decision

$$W_i(k) \quad \geq \quad K_1 \quad \forall i \in N_k^k \tag{19}$$

where $K_1 = Z_{LP_k^0}$. Then one of the two following assertions is true.

Either (LP_k^1) is infeasible and $Z_{LP_k^0}$ represents an upper bound on the optimal solution of (IP_k);

Or (LP_k^1) is feasible, and the solution is a convex combination of schedules with an identical score which is equal to $Z_{LP_k^1} = Z_{LP_k^0}$.

Proof

Since $Z_{LP_k^0}$ is the optimal value of (LP_k^0), it represents an upper bound on the value of the optimal solution of IP_k. Since LP_k^1 is more constrained than LP_k^0, the relation $Z_{LP_k^1} \leq Z_{LP_k^0}$ is true. Because of the addition of the resource constraint, the optimal solution of (LP_k^1) will be a convex combination of schedules whose scores are greater than or equal to $Z_{LP_k^0}$. Since the value of this convex combination must be less than or equal to $Z_{LP_k^0}$, then all the schedules of the convex combination have the same score which is equal to $Z_{LP_k^1} = Z_{LP_k^0}$. If such a convex combination does not exist, then (LP_k^1) is infeasible and $K_1 = Z_{LP_k^0}$ represents the best possible score that a schedule could have in the feasible solution of IP_k. Thus K_1 represents an upper bound on the optimal solution of (IP_k). ■

Let $Z_{INF_k^0}$ be the value of the column having the lowest score in the basis of the optimal solution of the first linear relaxation LP_k^0.

Proposition 2

Let (LP_k^{l+1}) be the linear relaxation problem obtained from (LP_k^l) after adding a new branching decision $W_i(k) \geq K_l \quad \forall i \in N_k^k$ and $Z_{LP_k^{l+1}}$ be the value of the optimal solution of (LP_k^{l+1}), where $\lfloor Z_{INF_k^0} \rfloor + 1 \leq K_l \leq Z_{LP_k^l}$. Then one of the two following assertions is true.

1. If (LP_k^{l+1}) is infeasible, then K_l represents a new upper bound on the optimal solution of (IP_k);
2. If (LP_k^{l+1}) is feasible, then $Z_{LP_k^{l+1}}$ represents a new upper bound on the optimal solution of (IP_k).

Proof

If it is impossible to find a convex combination of schedules whose scores are greater than or equal to K_l, then (LP_k^{l+1}) is infeasible and $K_l = Z_{LP_k^l}$ represents the best possible score that a schedule could have in the feasible solution of (IP_k). Therefore, K_l represents an upper bound on the optimal solution of (IP_k). On the other hand, if $l = 0$ and (LP_k^{l+1}) is feasible, then the following relation $K_0 \leq Z_{LP_k^1} \leq Z_{LP_k^0}$ is true, as explained in the proof of Proposition 1. Since $Z_{LP_k^1}$ represents the best possible score that a schedule could have in the feasible solution of IP_k, then $Z_{LP_k^1}$ represents a new upper bound on the optimal solution of (IP_k). Recursively, it can be shown that the relation $K_1 \leq Z_{LP_k^2} \leq Z_{LP_k^1}$ is also true, and finally that the relation $K_l \leq Z_{LP_k^{l+1}} \leq Z_{LP_k^l}$ is true. ■

Proposition 3

Let (LP_k^{l+1}) be the relaxed problem obtained from (LP_k^l) after adding the constraint

$$W_i(k) \quad \geq \quad K_l \quad \forall i \in N_k^k. \tag{20}$$

Let $Z_{INF_k^{l+1}}$ be the score of the variable having the lowest score in the convex combination of that solution. To make the search more efficient, the value of K_{l+1} in the next branching decision must fall into the following range:

$$\lfloor Z_{INF_k^{l+1}} \rfloor + 1 \leq K_{l+1} \leq Z_{LP_k^{l+1}}.$$

Proof

From Proposition 2, it has been proved that $Z_{LP_k^{l+1}}$ is an upper bound on the solution of IP_k. Any $K_{l+1} > Z_{LP_k^{l+1}}$ results in an infeasible solution. If the value of $K_{l+1} \leq Z_{INF_k^{l+1}}$, then the optimal solution obtained for LP_k^{l+1} remains feasible for LP_k^{l+2} and thus becomes an optimal solution; therefore, the branching decision would be of no utility. ■

4.4 Developing a branch and cut strategy

The analysis of the solutions obtained by the previous approach using cuts on several instances reveals that most of the time the optimal solution of the MIP_k problem is found in the interval:

$$\left[Z^0_{INF}, Z_{LP_k^0}\right].$$

Moreover, it has been noticed in Gamache *et al.* [8] that most of the time when the gap between Z^0_{INF} and $Z_{LP_k^0}$ was very large, the integrality gap was also very large. In these cases, sometime more than 10 cuts were needed to find the optimal solution to MIP_k.

Based upon these observations and the third proposition presented in the previous subsection, the following strategy has been used when applying the branching decisions:

$$K_0 = \lfloor Z^0_{INF} \rfloor + 1 < K_1 < K_2 < \cdots < K_l < K_{l+1} < \cdots \leq Z_{LP_k^0},$$

where $K_1 = \lfloor Z^1_{INF} \rfloor + 1$, $K_2 = \lfloor Z^2_{INF} \rfloor + 1$, etc.

If the value of an optimal solution is less than or equal to Z^0_{INF}, then the solution of the problem becomes infeasible with the addition of the constraint associated with the branching decision. In this case, the branch can be truncated and another branch is created from the parent node. In this new branch, the cut is used with Z^0_{INF} as a new upper bound on IP_k. This upper bound is much better than the one used before since $Z^0_{INF} < \lfloor Z^k_{LP} \rfloor$, which results in faster solution times. In the opposite case, i.e. if a convex combination of schedules having a score greater than or equal to Z^0_{INF} can be found, then the branch using the cut can be truncated. This propriety of eliminating one of the two branches at each level of the branching tree is very efficient and limits the search in the branch and bound tree to only one possibility.

The next section will present tests and results of the application of this new branching decision compared with the use of the previous cuts on some real problems.

5 Tests and results

The new branching strategy has been applied on real instances from Air Canada for the months of October and December. Six problems were tested. These problems deal with the construction of monthly work schedules for pilots for different fleets: Airbus A320, Airbus A340, Boeing 767, and DC9. These problems have been chosen because they were known for being difficult to solve and requiring the most number of cuts.

Table 1 shows the results obtained from both strategies: the cut and the branching decision. The first column indicates the name of the problem; the number in parenthesis indicates the number of pilots (iterations) whose schedule construction necessitates the use of cuts. Each of these iterations may necessitate more than one cut. The next two columns present the number of cuts used and the total CPU time needed during the use of these cuts. The same information is presented for the new branch and cut strategy in columns 4 and 5. Finally, the last two columns show the improvement of both the value of the solution (score of employee) and the solution time between the new branch and cut strategy and the cut. The tests were conducted on a Sun Ultra 10.

Problem	**Cut**		**Branch and cut**		**Improvement(%)**	
(Pilots using cuts)	Number	CPU(s)	Number	CPU(s)	(CPU)	Solution
OCT-340-ca(4)	12	63.8	11	27.7	57	17
OCT-767-ca(8)	21	411.45	16	278.7	33	71
OCT-dc9-ca(4)	7	32.4	4	4	88	0
OCT-320-ca(11)	51	1301	39	639	51	101
DEC-767-ca(8)	68	15407	12	2103	87	7
DEC-320-ca(17)	130	15312	26	4743	69	13

Table 1: Comparison between the cut and the branching strategy (October 2000)

One can notice that a clear improvement on the solution time (between 33% and 88%) has been obtained when using the new branch and cut strategy. In addition to the reduction of the solution time per iteration, the number of nodes in the branch and cut tree has also decreased. This improvement is mostly due to the presence of large integrality gaps in some of these problems and also because the optimal solution is very close, in most of the cases, to the value Z^0_{INF}. Moreover, the branch and cut strategy results in an improvement of the solution for some pilots compared to the same solution obtained when using the cuts. Since the resource used for the cut creates a huge number of non-dominated labels, a heuristic approach has been used in order to select only a subset of these labels. The new approach based upon branch and cut decisions involves the use of a resource that permits an efficient dominance. Since the number of labels was much smaller when using the branching decision strategy, all of them were kept during the solution process of the subproblem which resulted in the generation of schedules having a better score.

However, the improvement of the solution makes the comparison of the solution time more difficult to interpret. Indeed, a better solution for a pilot may influence the solution time for the search of a feasible solution for the most junior employees. The score for more junior employees may differ and a comparison between the two solution processes depends on a supplementary factor. In order to make the comparison possible, two series of tests have been conducted using the data for the month of December for the Airbus A320 fleet because this problem is the one having the biggest number of pilots.

For the first series of tests, 24 problems needing at least 1 cut each were used. In each of these problems, the solution process was stopped after the construction of the schedule corresponding to the iteration where previously the cut was needed and both approaches were compared. Table 2 presents the results of this comparison between the two strategies. One can notice that the new strategy is more efficient than the strategy using cuts. The number of branch and cut decisions is smaller than the number of cuts and the time spent for each branching decision is also smaller that the time spent for each cut. Moreover, it is also interesting to indicate the percentage of the integrality gap in order to show the efficiency of the branch and cut strategy in the cases where huge gaps were encountered. Table 2 presents the mean and also the maximum value of these gaps. One can notice that the branch and cut strategy is very efficient when huge integrality gaps are observed. In several cases, a gap greater or equal to 90% has been reduced using only one branching decision.

Number	**Cut**		**Branch and cut**		**Improvement**	**Gap (%)**	
of problems	Number	CPU(s)	Number	CPU (s)	% (CPU)	Mean	Max
9	20	2253	10	1196	47	2.5	6.1
5	39	4604	5	458	90	64.9	99.2
5	22	1464	5	378	74	71.2	93.4
5	8	792	5	512	36	62.4	99.1

Table 2: Comparison between the cut and the branching strategy: Airbus 320

Indeed, for some of the problems that have been tested, the number of branching decisions compared to the number of cuts has been reduced by a factor of almost 8. In most of these problems, the value of the optimal solution is equal to the value of the column having the smallest score in the basis of the optimal solution of the linear relaxation before the addition of branching decisions.

In the second series of tests, for each iteration k where a cut was used in the old solution approach, the $k-1$ schedules obtained previously were kept. The new approach using the branch and cut decision started directly at iteration k with $k-1$ schedules (found with the other solution approach) already assigned to the $k-1$ most senior employees. This process was repeated every time a cut was used, which produced as many problems as there were cuts. This way, a specific comparison between the two strategies is possible each time a cut is used. Table 3 shows the results of both strategies (cut and branching) on 9 different problems where at least 5 pilots needed a cut.

Number	**Cut**		**Branch and cut**		**Improvement**
of problems	Number	CPU(s)	Number	CPU(s)	% (CPU)
5	39	4604	5	458	90
5	40	4646	5	529	89
5	10	409	5	210	49
5	10	983	6	725	28
5	8	441	5	234	47
5	21	1259	5	521	59
5	8	486	7	316	35
5	8	844	6	429	49
5	39	4489	5	436	90

Table 3: Comparison between the cut and the branching strategy: Airbus 320

Once more, it can be seen that the use of the branching strategy instead of the cut improves the solution of the longest path problem with resource constraints when the solution of the relaxation of the linear program is a convex combination of schedules having different scores.

6 Other applications

The branch and cut strategy described in Section 4 can be applied to several other problems providing that they can be formulated as the linear program described by equations (1) to (4) and solved by the column generation technique which is imbedded in a branch-and-bound algorithm, and that a bound on some subproblems can be obtained from Z_{LP} of the master problem. This strategy is very effective in improving the bound given by the linear program relaxation as long as the solution of this relaxation is a combination of columns having different costs. Very large improvements may be obtained with few branch and cut decisions.

6.1 Problems having only one subproblem

The branch and cut strategy can be applied in a column generation context having only one subproblem. One typical example is the shortest path problem with additional linear constraints (Minoux and Gondran, 1979). This problem can be solved by column generation with the additional constraints in the master problem and a single shortest path subproblem. The branch and cut decision will add a single resource constraint to the subproblem. Such a problem is easy to solve particularly when the resource is positively correlated with the cost function to minimize.

In this case, the solution strategy is similar to the one presented for the PBS problem. Similar solution time reductions are expected.

6.2 Problems having several subproblems

In the PBS context, the branch and cut strategy has been applied efficiently in a column generation approach having several subproblems. The particularity of this problem is that at each iteration, the master problem provides a bound for the subproblems. This is due to the linear relaxation of the integrality constraints of the master problem $Z_{LP}(MP)$ but also to its formulation. In such problems, the following relations exist:

$$Z^1_{SP} \leq Z_{MP} \tag{21}$$

$$Z^k_{SP} = 0 \;\; \forall k \geq 2 \tag{22}$$

where Z_{MP} is the optimal solution of the restricted master problem and Z^k_{SP} the values of the columns generated by subproblem k.

This relation allows us to use Z^1_{SP} in branch and cut decisions. Other applications have the same type of formulation where there is only one subproblem having costs. Other applications have the same type of formulation. This is the case for the travelling salesman problem. Wong (1980), Langevin *et al.* (1988) and Loulou (1988) have presented a formulation of this problem which can be separated into a master problem and $2k$-1 subproblems, where k represents the number of nodes in the graph. The first subproblem consists of finding for the travelling salesman a route visiting all nodes starting from node 1 (the depot) and returning to that node. This route is found by solving a shortest path problem. Two subproblems (SP^+_i and SP^-_i) are associated with each node i of the graph, where $i \neq 1$. In SP^+_i, one unit of commodity flow (C^+_i) is transported from node 1 to node i, while in SP^-_i, one unit of commodity flow (C^-_i) leaves node i and returns to node 1. Flow conservation constraints at each node are respected in each subproblem. In this formulation, the objective function of the master problem depends only on the variables of the first subproblem. The master problem consists of minimizing the route of the travelling salesman while imposing two conditions: (1) the route must visit all the nodes of the graph and (2) for each arc of the graph, the values of variables associated with the commodities C_i^+ and C_i^-, $i = 2, .., k$, must be less than or equal to that of the variable of the route.

6.3 Minmax problems

The proposed branch and cut method can also be applied to $Min\ C_{max}$ optimization problems. A first application is the project management problem with time windows and resources constraints. This problem is called in the literature "*Generalized Resource Constrained Project Scheduling Problem*". The objective function of this problem can be formulated as follow:

$$Z_{MP} = \max_k \ Z_{SP_k}$$

where Z_{SP_k} is the value of the subproblem k's solution. This class of problems can be solved by column generation, where the resource constraints are included into the master problems, and both the precedence constraints and the time windows are taken into account in the subproblems.

The technique of cuts on the subproblem has also been used by Milon *et al.* (1999) in the context of minimizing the duration of project management problems (Min C_{max}). These authors have used the solution of the linear relaxation as a lower bound in order to generate columns with a cost greater or equal to Z_{LP}^*. This is done by tightening the windows of the last activity of the project in order to generate columns with lower cost. This lower bound is taken into account by adding a delay on the last task without affecting the time windows of other activities. Thus, the columns with a cost lower than Z_{LP}^* are penalized without being forbidden in the list of generated columns. The new branch and cut strategy could be used to solve this problem by imposing the constraint that the subproblem generate columns having a cost less than a given value.

The branch and cut strategy can be used in a second application: the scheduling problem. The parallel machine problem with C_{max} as an objective has been solved by Martello *et al.* (1992) by using lower bounds based on the Lagrangian relaxation and on the annexation of these bounds. The relation between C_{max} and the bound obtained from relaxation is used to compute cuts on the objective function. They have been used in an enumeration and elimination approach which was found to be more efficient than all the algorithms found in the literature. This problem can be solved by column generation where integrality constraints are relaxed. The new branch and cut strategy can also be used in this context in order to find an integer solution when the linear relaxation is a convex combination of columns having different scores. Gélinas (1997) has solved the Job-Shop Scheduling

Problem with resource constraints by using column generation. In her formulation, the master problem chooses a convex combination of schedules generated in such manner as to respect the preceding constraints by minimizing the completion time of the machines. In such problem, the resource constraints are transferred to the subproblems. For each machine, there is a subproblem representing a scheduling problem with time windows. We have the following relation ($Z_{SP_j} \leq Z_{MP}$) between the objective of a subproblem and the objective of the master problem and the branching on Z_{MP} induces constraints on Z_{SP_j} for all subproblems.

7 Conclusion

In this paper, a new branch and cut strategy has been presented. This new strategy, which is applied on subproblems during the solution of a large-scale linear program by a column generation technique, has been used to improve the solution time of large-scale problems having a particular structure. The branch and cut strategy has been applied on preferential bidding problems that are encountered in the airline industry. These tests have shown that the new branch and cut strategy provides important improvements compared to a similar strategy based on cuts (Gamache *et al.*, 1998b). First, it improves the solution time because the resource used in the branch and cut strategy is positively correlated with other resources of the subproblem which was not the case when using the cuts. Secondly, each branch and cut decision is very efficient because it allows either the generation of an optimal integer solution or the production of an new upper bound on the value of the optimal solution of MIP_k. Thirdly, the new approach has produced better schedules for many pilots on different instances from Air Canada, which invalidates the upper bound produced by the old cut due to a non optimal solution of the subproblem.

Moreover, it has been shown that the branch and cut strategy developed for the PBS problem can also be applied to different problems having the same particular structure.

References

[1] ANTOSIK, J.L.(1978). Automatic Monthly Crew Assignment, a New Approach. 1978 AGIFORS Symposium Proceeding, 18, 369-402.

[2] BUHR, J.(1978). Four Methods for Monthly Crew Assignment - A Comparison of Efficiency. 1978 AGIFORS Symposium Proceeding, 18, 403-430.

[3] BYRNE, J. (1988). A Preferential Bidding System for Technical Aircrew. 1988 AGIFORS Symposium Proceeding, 28, 87-99.

[4] CPLEX Reference Manual (1992). Using the Cplex Callable Library and CPLEX Mixed Integer Library. Cplex Optimisation, Inc., Incline Village, NV 89451-9436, U.S.A.

[5] J. DESROSIERS, A.LASRY, D. McINNIS, M.M. SOLOMON et F. SOUMIS, “ALTITUDE: The Airline Operations Management System at Air Transat,” Les Cahiers du GERAD G-95-23, École des Hautes Études Commerciales, Montréal, Canada, 1995.

[6] FEDERICI, F. et PASCHINA, D. (1990). Automated Rostering Model. 1990 AGIFORS Symposium Proceedings, 12.

[7] GAMACHE, M., F. SOUMIS.(1998). A method for optimality solving the rostering problem. G. Yu, ed. Operations Research in the Airline Industry, Kluwer, Boston. 124-157.

[8] GAMACHE, M., F. SOUMIS, J. DESROSIERS, D. VILLENEUVE, And E. GÉLINAS. (1998). The Preferential Bidding System at Air Canada. Transportation Science. V32 246-255.

[9] GAMACHE, M., F. SOUMIS, G. MARQUIS, and J. DESROSIERS. (1999). A Column Generation Approach for Large-Scale Aircrew Rostering Problems. Operations Research. 247-262.

[10] GÉLINAS, S. (1997) Problèmes d’ordonnancement. Ph. D. Thesis. École Polytechnique de Montréal. pp 197.

[11] GIAFFERRI, C., J.P. HAMON, J. G. LENGLINE. (1982). Automatic Monthly Assignment of Mesdium-Haul Cabin Crew. 1982 AGIFORS Symposium Proceeding, 22, 69-95.

[12] GLANERT, W. (1984). A Timetable Approach to the Assignment of Pilots to Rotations. 1984 AGIFORS Symposium Proceeding, 24, 369-391.

[13] GONTIER, T. (1985). Longhaul Cabin Crew Assignment. 1985 AGIFORS Symposium Proceeding, 25, 44-66.

[14] LANGEVIN, A., DESROSIERS, J. (1988). Classification of Traveling Salesman Problem Formulations. Operations Res. Lett. 2 (1990) 393-410.

[15] LOULOU, R.J. (1988). On multicommodity flow formulation for the TSP, working paper, McGill University, Montréal, Canada.

[16] Martello, S.,Soumis,F.,Toth, P. (1997) Exact and Approximation algorithms for Makespan Minimization on Unrelated Parallel Machines, Descried Applied Mathematics, May 30^{th} 1997, volume 75, issue 2, pages 169-188.

[17] MARCHETTINI, F. (1980). Automatic Monthly Cabin Crew Rostering Procedure. 1980 AGIFORS Symposium Proceeding, 20, 23-59.

[18] MAYER, M. (1980). Monthly Computerized Crew Assignment. 1980 AGIFORS Symposium Proceeding, 20, 93-124.

[19] MILON, O., SOUMIS, F., Tachefine, B. (1999) Gestion de Projet avec Contraintes de Ressources. Rapport de Présentation de Maitrise, École Polytechnique de Montréal.

[20] MINOUX, M., GONDRAN, M. (1979). Graphes et Algorithmes, Ed. Eyrolles, 1979.

[21] MOORE, R., EVANS, J. and NGO, H. (1978). Computerized Tailored Blocking. 1978 AGIFORS Symposium Proceeding, 18, 343-361.

[22] MORI, A. (1988). Cockpit Crew Scheduling System by AI. 1988 AGIFORS Symposium Proceeding, 28, 74-86.

[23] NICOLETTI, B.(1975). Automatic Crew Rostering. Transportation Science, 9, 33-42.

[24] RUSSELL, D.J. (1989). Development of an Automated Crew Bid Generation System. INTERFACES, 19, 44-51.

[25] RYAN, D.M. et FALKNER, J.C (1988). On the Integer Properties of Scheduling Set Partionning Models. European Journal of the Operational Research, 35, 442-456.

[26] RYAN, D.M.(1992). The Solution of Massive Generalized Set Partitioning Problems in Air Crew Rostering. Operations Research, 45, 649-661.

[27] RYAN, D.M, Day, R.D (1997). Flight Attendant Rostering for Short-Haul Airline Operations. Journal of the Operational Research Society, 43, 459-467.

[28] SARRA, D.(1988). The Automatic Assignment Model. 1988 AGIFORS Symposium Proceeding, 28, 23-37.

[29] SCHMIDT, W.R et HOSSEINI, J.(1984). Preferential Schedule Assignments for Airline Crew Schedulling.

[30] TINGLEY, G.A.(1979). Still Another Solution Method for the Monthly Aircrew Assignment Problem, 1979 AGIFORS Symposium Proceeding, 19, 143-203..

[31] WILSON, B.J. (1981). BA's Regular O.R. Crew Planning Models for the 1980's. 1981 AGIFORS Symposium Proceeding, 21, 257-270.

[32] WONG,R.T. (1980). Integer Programming Formulation of the Travelling Salesman Problem, Proceedings of the IEEE International Conference of Circuits and Computers, 149-152.