AN EXACT ALGORITHM FOR IP COLUMN GENERATION

François Vanderbeck
Management studies group, Cambridge University
Trumpington st., Cambridge CB2 1AG, UK

and

Laurence A. Wolsey
CORE, Université Catholique de Louvain
Voie du Roman Pays 34, 1348 LLN, Belgium

April 1994 (revised August 1995)

ABSTRACT

An exact column generation algorithm for integer programs with a large (implicit) number of columns is presented. The family of problems that can be treated includes not only standard partitioning problems such as bin packing and certain vehicle routing problems in which the columns generated have $0-1$ components and a right hand side vector of 1's, but also the cutting stock problem in which the columns and right hand side are nonnegative integer vectors. We develop a combined branching and subproblem modification scheme that generalizes existing approaches, and also describe the use of lower bounds to reduce tailing-off effects.

Keywords: Integer Programming, Column Generation

This text presents research results of the Belgian programme on interuniversity poles of attraction, PAI contract no.26, initiated by the Belgian state, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by its authors. This research was supported in part by Science Program SC1-CT91-620 of the EEC, and F. Vanderbeck was partially supported by a doctoral fellowship from the Centre Interuniversitaire d'Etudes Doctorales dans les Sciences du Management (CIM).

1 Introduction

Column generation formulations of integer programs have been proposed and discussed for several decades. Starting from the pioneering work of Gilmore and Gomory [5,6] on the cutting stock problem, Minoux [12] shows how several important combinatorial optimization problems can be reformulated and tackled by column generation, numerous applications in the routing/distribution area are covered in a recent survey of Desrosiers et al. [3], and recently Hansen et al. [7], Savelsbergh [14], Vance et al. [16] among others have shown different ways in which column generation and branch and bound can be combined to obtain guaranteed optimal solutions. Barnhart et al. [1] survey some of the recent work in this area.

The problem to be solved is the Integer Program (Master):

$$
\begin{array}{cc}
 & z = \min \sum_{q \in Q} c_q \lambda_q \\
(P) & \sum_{q \in Q} q \lambda_q = b \\
 & \sum_{q \in Q} \lambda_q \leq K \\
 & \sum_{q \in Q} \lambda_q \geq L \\
 & \lambda_q \geq 0 \text{ and integer }, \ q \in Q
\end{array}
$$

where $q, b \in Z_+^m$, W is a set defining the set of feasible columns and their associated costs, i.e. $Q = \{x : (x, w) \in W\} \subset Z_+^m$, $c_q = \min_w \{ex + fw : x = q, (x, w) \in W\}$ is the minimum cost solution associated with column $q \in Q$, where $(e, f) \in R^m \times R^p$ is the cost vector and $w \in R^p$ represents auxiliary variables that may be real or integer. The variable λ_q (indexed by a vector $q \in Z_+^m$) is the number of times the vector $q \in Q$ is chosen, and L and K are lower and upper bounds on the total number of columns. We set $M = \{1, \ldots, m\}$.

Typically the Master problem has such a large number of columns that they cannot be written out explicitly. Thus in practice one works with a linear programming relaxation involving only a small subset of the columns of Q. New columns are generated as needed when their reduced costs are negative and they are therefore candidates to improve the objective function. Using optimal dual variables $(\pi, \mu, \nu) \in R^m \times R_- \times R_+$ from this restricted linear programming relaxation, a pricing subproblem is solved exactly or heuristically to generate a column of negative reduced cost. This pricing

subproblem takes the form:

$$\zeta = \min_{x,w}\{ex + fw - \pi x : (x,w) \in W\} - \mu - \nu,$$

and a solution (x,w) leads to a new column with $q = x, c_q = ex + fw$.

Problem (P) contains many well-known models as special cases. The cutting stock problem is the case in which $q \in Q$ represents a feasible cutting pattern in which q_i pieces of length a_i are cut from a piece of length A. Here one can take $W = \{(x,w) \in Z_+^m \times \{0,1\} : \sum_{i=1}^m a_i x_i \leq Aw\}$ with $e = 0$ and $f = 1$, so $c_q = w$ for all $q \in Q$. The right hand side b is the demand vector with b_i indicating how many pieces of length a_i must be cut in total.

The single depot capacitated vehicle routing problem involves a graph $G = (V,E)$, κ vehicles of capacity C, depot node 0, client demands d_i for $i \in V\backslash\{0\}$, and edge lengths f_e for $e \in E$. Taking q to be the characteristic vector of client subsets $S^q \subseteq V \setminus \{0\}$ visited on a feasible tour, we can take $W = \{(x,w) \in \{0,1\}^m \times \{0,1\}^{|E|} : \sum_{e\in\delta(i)} w_e = 2x_i \text{ for } i \in V \setminus \{0\}, \sum_{e\in\delta(0)} w_e = 2, \sum_{e\in\delta(S)} w_e \geq 2x_i \text{ for all } i, S \text{ with } 0 \notin S, i \in S, \sum_{i\in M} d_i x_i \leq C\}$ where $\delta(S) = \{(i,j) : i \in S, j \in V \setminus S\}$ and $\delta(i) = \delta(\{i\})$ so that edges w_e form a single subtour including the depot node 0. Also $e = 0$, and c_q is the length of the shortest tour through the nodes $S^q \cup \{0\}$. Here b is a vector of 1's indicating that each client is visited once, and $L = 1$ and $K = \kappa$ represent bounds on the number of vehicles used.

It is also possible to treat problems where columns can be generated from different subsets, in which case we talk of different (as opposed to identical) subproblems. For instance the single machine multi-item lot-sizing model with κ items, n periods and demands d_t^k and production capacities C_t^k for item k in period t, where only one item can be produced in each period, is a special case in which $q \in R^{n+\kappa}$ and $q = \binom{x^k}{e^k}$ for some k, where $W^k = \{(x^k, e^k, y^k, s^k) \in \{0,1\}^n \times \{0,1\}^\kappa \times R_+^n \times R_+^n : s_{t-1}^k + y_t^k = d_t^k + s_t^k, y_t^k \leq C_t^k x_t^k \text{ for all } t\}$ representing the feasible production region for item k where $w = (y^k, s^k)$ are the production and stock variables respectively, x^k is the characteristic vector of the production periods for item k, and e^k is the kth unit vector. Here $b_i = 1$ for all i indicates that exactly one item is produced in each period, and that exactly one production plan must be chosen for each item. The cardinality constraints with $L = K = \kappa$ are redundant.

Problems of the form (P) sometimes arise directly, but are also often obtained by applying Dantzig-Wolfe decomposition (Dantzig and Wolfe [2])

to some different initial formulation. Thus Gilmore and Gomory formulated the cutting stock formulation directly in the form (P), but it is not the usual starting formulation for the multi-item lot-sizing problem. See Magnanti and Wolsey [11] for a discussion of various other models leading to formulations of the form (P).

Here we present an algorithm that solves the problem (P) exactly. It has been largely motivated by problems with identical subproblems, and in which $b, q \in Z_+^m$ in contrast to most previous work in which $q \in \{0,1\}^m$ and $b_i = 1$ for all $i \in M$. To our knowledge no satisfactory general branching scheme has been proposed for such problems, see the recent survey [1]. Though the scheme we propose is largely "conceptual", the algorithm can be implemented using a modular mixed integer programming system such as CPLEX, MINTO, OSL or XPRESS if an explicit mixed integer formulation of the set W defining the set of feasible columns is provided. The algorithm may be suitable for certain complicated problems for which column generation appears to be an appropriate approach, but in which the subproblem is still a difficult integer program. This is in contrast to most specific applications treated to date, where the subproblems are relatively easy problems that are solved rapidly by special purpose algorithms or heuristics. The desire for simplicity and speed is motivated by the fact that the subproblem needs to be solved a large number of times.

The contents of the paper are as follows. Section 2 contains a description of the algorithm, in particular a valid branching scheme, a corresponding method to modify the subproblems after branching, and verification that it can be implemented. In Section 3 we discuss the separate issue of how lower bounds can be used to improve the performance of the basic algorithm. In Section 4 we discuss how various steps of the algorithm simplify when the feasible columns are only 0-1 vectors, so that the problem is one of set partitioning where each item must be chosen b_i times. We then further specialize by taking $b_i = 1$ for all i which gives the problem generally treated in earlier work.

In Section 5 we present some computational experience from two problems in order to show first that the "conceptual" algorithm and branching scheme presented in Section 2 can be implemented as described to solve a general class of cutting stock problems, and secondly to report briefly on the effectiveness of the early termination rules of Section 3 on a class of telecommunications problems. Finally we discuss some outstanding issues.

2 The IP Column Generation Algorithm

We describe a branch and bound algorithm for (P) in which we show that all fractional master solutions can be eliminated by branching on general upper (lower) bound constraints of the form

$$\sum_{q \in Q(q^j)} \lambda_q \leq K^j \quad (\geq L^j),$$

where $Q(p) = \{q \in Q : q \geq p\}$ for $p \in Z_+^m$.

Thus at each node u of an enumeration tree, we associate a problem P^u that is characterised by cardinality constraints arising from branching. Let G^u and H^u be the index sets of branching constraints at node u consisting of pairs (q^j, K^j) and pairs (q^j, L^j) respectively.

The problem at node u then takes the form:

$$(P^u) \qquad \begin{array}{c} z^u = \min \sum_{q \in Q} c_q \lambda_q \\ \sum_{q \in Q} q \lambda_q = b \\ \sum_{q \in Q(q^j)} \lambda_q \leq K^j \quad j \in G^u \\ \sum_{q \in Q(q^j)} \lambda_q \geq L^j \quad j \in H^u \\ \lambda_q \geq 0 \text{ and integer} \quad q \in Q. \end{array}$$

Note that the initial cardinality constraints $L \leq \sum_{q \in Q} \lambda_q \leq K$ present at the root node are also present at node u as $Q = Q(0), 0 \in G^u$ represents the pair $(0, K)$, and $0 \in H^u$ represents the pair $(0, L)$.

With (P^u) we associate a restricted linear program over a subset $\tilde{Q} \subseteq Q$ of the columns:

$$(LP^u(\tilde{Q})) \qquad \begin{array}{c} z^u_{LP}(\tilde{Q}) = \min \sum_{q \in \tilde{Q}} c_q \lambda_q \\ \sum_{q \in \tilde{Q}} q \lambda_q = b \\ \sum_{q \in \tilde{Q}(q^j)} \lambda_q \leq K^j \quad j \in G^u \\ \sum_{q \in \tilde{Q}(q^j)} \lambda_q \geq L^j \quad j \in H^u \\ \lambda_q \geq 0 \; q \in \tilde{Q} \end{array}$$

where $\tilde{Q}^j = Q(q^j) \cap \tilde{Q}$, and $(\pi, \mu, \nu) \in R^m \times R_-^{|G^u|} \times R_+^{|H^u|}$ are optimal dual variables.

With $(LP^u(\tilde{Q}))$ we associate the subproblem designed to test whether any column $q \in Q \setminus \tilde{Q}$ has a negative reduced cost. To calculate the reduced cost of a column $q \in Q$, we need to include the values of the dual variables μ_j, ν_j depending on whether $q \in Q(q^j)$ or not. To do this we explicitly introduce a 0-1 variable z^j in the subproblem which takes the value 1 if $q \in Q(q^j)$ and 0 otherwise. The resulting subproblem is:

$$
\begin{array}{rl}
\zeta^{\pi,\mu,\nu} = & \min(e-\pi)x + fw - \mu_0 - \nu_0 - \sum_{j \in G_0^u} \mu_j z^j - \sum_{j \in H_0^u} \nu_j z^j \\
(SP^u(\pi,\mu,\nu)) & (x,w) \in W \\
& (x,z^j) \in X_{q^j} \quad j \in G_0^u \cup H_0^u
\end{array}
$$

where $X_{q^j} = \{(x,z^j) : z^j = 1 \text{ if } x \geq q^j \text{ and } z^j = 0 \text{ otherwise}\}$, $G_0^u = G^u \setminus \{0\}$ and $H_0^u = H^u \setminus \{0\}$.

Now at each node u, a standard column generation algorithm is applied to solve problem $LP^u(Q)$. At each iteration the restricted LP Master Problem $LP^u(\tilde{Q})$ is solved giving an upper bound $z_{LP}^u(\tilde{Q})$ on the value of the linear programming relaxation $z_{LP}^u(Q)$. Taking an optimal dual solution (π,μ,ν), the subproblem $SP^u(\pi,\mu,\nu)$ is then solved. If $\zeta^{\pi,\mu,\nu} < 0$ and (x,w) is an optimal solution, a new column $q = x$ with $c_q = ex + fw$ is found, $\tilde{Q}$ is updated and a new iteration begins. If $\zeta^{\pi,\mu,\nu} = 0$, $LP^u(Q)$ is solved, and either node u is pruned by standard arguments, or branching takes place.

In the latter case, let λ^* be an optimal solution of $LP^u(Q)$. λ^* is not integral.

SEPARATION. Choose q^* such that $\sum_{q \in Q(q*)} \lambda_q^* = \alpha$ is fractional.

BRANCHING. Add two problems to the node list consisting of problem (P^u) plus the single constraint:

$$\sum_{q \in Q(q^*)} \lambda_q \leq \lfloor \alpha \rfloor$$

and (P^u) plus the single constraint:

$$\sum_{q \in Q(q^*)} \lambda_q \geq \lceil \alpha \rceil$$

respectively.

To establish that this algorithm can be implemented, we need to show that an appropriate vector q^* can be found for the SEPARATION step, and that the subproblems at each node can be formulated as mixed integer programming problems.

Proposition 1 *Given a feasible solution λ^* of (P^u) that is not integral, there exists a vector $q^* \in Z^m$ such that $\sum_{q \in Q(q^*)} \lambda_q^*$ is fractional.*

Proof. Take q^* to be any maximal (undominated) element of the set $\{q \in Q : \lambda_q^* \notin Z_+^1\}$. Such an element always exists as Q is a finite set. Then $\sum_{q \in Q(q^*)} \lambda_q^* = \lambda_{q^*}^* \notin Z_+^1$. ∎

Now we need to show that sets of the form $X_p = \{(x,z) : z = 1$ if $x \geq p, z = 0$ otherwise $\}$ with $p \in Z_+^m$ can be formulated as MIPs. Let $P^+ = \{i \in M : p_i > 0\}$ be the support of p. We introduce variables η^i for $i \in P^+$ where $\eta^i = 1$ if $x_i \geq p_i$ and $\eta^i = 0$ if $x_i < p_i$.

Proposition 2 *X_p is the set of (x,z) satisfying*

$$p_i \eta^i \leq x_i \leq (p_i - 1) + (b_i - p_i + 1)\eta^i \quad i \in P^+$$

$$z \leq \eta^i \quad i \in P^+$$

$$z \geq 1 - \sum_{i \in P^+} (1 - \eta^i)$$

$$x \in Z_+^m, z \in \{0,1\}, \eta^i \in \{0,1\} \quad i \in P^+.$$

Note that in practice it is desirable to choose a branching vector q^* with as few positive components as possible, so as to minimise the changes in the subproblem.

As Q, the number of vectors q^j and scalars L^j and $K^j (\leq \sum_{i=1}^m b_i)$ are finite, and the problems (P^u) are more restricted at each level of the enumeration tree, it is easily verified that the integer programming column generation algorithm terminates finitely.

We terminate this section with some observations showing how the branching constraints, and additional subproblem constraints, simplify when K^j and L^j take their minimum and maximum values respectively.

Observation 1. If $K^j = 0$, then one can set $\lambda_q = 0$ for all $q \in Q(q^j)$ in (P^u), and set $z^j = 0$ in the subproblem. The resulting set X_{q^j} reduces to x satisfying:

$$x_i \leq (q_i^j - 1) + (b_i - q_i^j + 1)\eta^i \; i \in P^+$$
$$\sum_{i \in P^+} (1 - \eta^i) \geq 1$$
$$x \in Z_+^m, \eta^i \in \{0,1\} \; i \in P^+.$$

where here $P^+ = \{i \in M : q_i^j > 0\}$.

We can also bound K^j.

Observation 2. $K^j \leq \min_{i:q_i^j \geq 1} \lfloor \frac{b_i}{q_i^j} \rfloor$ as $q_i^j \sum_{q \in Q(q^j)} \lambda_q \leq \sum_{q \in Q(q^j)} q_i \lambda_q \leq \sum_{q \in Q} q_i \lambda_q = b_i$ for all $i \in M$ such that $q_i^j \geq 1$.

Now suppose that for some $j \in H^u$, L^j takes its maximum possible value as defined in Observation 2, i.e. $L^j = \min_{i:q_i^j \geq 1} \lfloor \frac{b_i}{q_i^j} \rfloor$. If $\min_{i:q_i^j \geq 1} \lfloor \frac{b_i}{q_i^j} \rfloor = \min_{i:q_i^j \geq 1} \frac{b_i}{q_i^j}$, then $I = \{i : q_i^j \geq 1 \text{ and } \frac{b_i}{q_i^j} = L^j\} \neq \phi$.

Observation 3. If $L^j = \min_{i:q_i^j \geq 1} \frac{b_i}{q_i^j}$, one can set $\lambda_q = 0$ for all $q \in Q \backslash Q(q^j)$ satisfying $\sum_{i \in I} q_i > 0$, and also set $\lambda_q = 0$ for all $q \in Q(q^j)$ with $q_i > q_i^j$ for some $i \in I$. The resulting set X_{q^j} reduces to (x, z^j) satisfying

$$x_i = q_i^j z^j \quad i \in I$$
$$x_i \geq q_i^j z^j \quad i \in P^+ \setminus I$$
$$z^j \in \{0,1\}.$$

3 Lower Bounds for Early Termination of Column Generation

In practice, one of the well-known difficulties with column generation is the "tailing-off effect", namely the large number of iterations often needed to prove LP optimality. Potentially this can now happen at each node of the branch and bound tree, and what is more the subproblem to be solved at

each iteration is now typically a difficult integer program. In addition the subproblem becomes considerably harder to solve as the dual variables converge to the optimal dual values for z^u_{LP}. This means that it is of crucial importance to control the tailing-off effect.

Here we show how the basic algorithm can be adapted so as to partially tackle this drawback. First we need to obtain as tight a lower bound as possible on $z^u_{LP}(Q)$ at each iteration.

Proposition 3 *Given* $(\pi, \mu, \nu) \in R^m \times R_-^{|G^u|} \times R_+^{|H^u|}$, $z^u_{LP}(Q) \geq LB^{\pi,\mu,\nu} = \min\{K^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0), L^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0)\} + \sum_{i\in M} \pi_i b_i + \sum_{j\in G^u_0} \mu_j K^j + \sum_{j\in H^u_0} \nu_j L^j)$.

Proof. Dualizing all the constraints of $LP^u(Q)$ except for the initial cardinality constraints $0 \in G^u \cap H^u$ gives:

$$z^u_{LP}(Q) \geq \sum_{i\in M} \pi_i b_i + \sum_{j\in G^u_0} \mu_j K^j + \sum_{j\in H^u_0} \nu_j L^j$$
$$+ \min \sum_{q\in Q} [c_q - \sum_{i\in M} \pi_i q_i - \sum_{j\in G^u_0 : q\geq q^j} \mu_j - \sum_{j\in H^u_0 : q\geq q^j} \nu_j)]\lambda_q$$

subject to

$$L^0 \leq \sum_{q\in Q} \lambda_q \leq K^0$$
$$\lambda_q \geq 0 \ q \in Q.$$

The optimal solution of a two-sided cardinality knapsack problem:

$$\min\{\sum_{q\in Q} \gamma_q \lambda_q : L^0 \leq \sum_{q\in Q} \lambda_q \leq K^0, \ \lambda_q \geq 0 \ \ q \in Q\}$$

is $\min\{K^0\gamma^*, L^0\gamma^*\}$ where $\gamma^* = \min_q \gamma_q$. The claim now follows from the definition of $\zeta^{\pi,\mu,\nu}$. ∎

Note that when $\zeta^{\pi,\mu,\nu} = 0$, $z^u_{LP}(Q) = LB^{\pi,\mu,\nu}$. Lasdon[10] and Farley[4] have also proposed lower bounds. The Lagrangian bound proposed above is a minor strengthening of that of Lasdon. Farley's is different in that it is based on rescaling of the dual variables π, and may often require the solution of an additional subproblem.

To make precise the potential value of a good lower bound, we suppose that the optimal value z of (P) is integral as in many applications, and that a best feasible solution of value z^{INC} has been found, as well as a best possible lower bound $\underline{z}^u_{LP}$ on $z^u_{LP}(Q)$, namely the largest value of $LB^{\pi,\mu,\nu}$ over the previous iterations. This leads to:

Observation 4. If $\lceil \underline{z}^u_{LP} \rceil \geq z^u_{LP}(\tilde{Q})$, then $z^u \geq \lceil z^u_{LP}(\tilde{Q}) \rceil = \lceil z^u_{LP}(Q) \rceil$, so further work at node u will not produce a better lower bound, and column generation at the node can be terminated.

Observation 5. If $\lceil \underline{z}^u_{LP} \rceil \geq z^{INC}$, then $z^u \geq z^{INC}$, and node u can be pruned.

Observations 4 and 5 can be used once the subproblem at an iteration has been solved. What can be concluded *a priori* before solving the subproblem? Let $\rho_1 = \max\{\frac{\delta_1}{K^0}, \frac{\delta_1}{L^0}\} - \mu_0 - \nu_0$ where

$$\delta_1 = \lceil z^u_{LP}(\tilde{Q}) \rceil - 1 - \sum_i \pi_i b_i - \sum_{j \in G^u_0} \mu_j K^j - \sum_{j \in H^u_0} \nu_j L^j$$

and $\rho_2 = \max\{\frac{\delta_2}{K^0}, \frac{\delta_2}{L^0}\} - \mu_0 - \nu_0$, where

$$\delta_2 = z^{INC} - 1 - \sum_i \pi_i b_i - \sum_{j \in G^u_0} \mu_j K^j - \sum_{j \in H^u_0} \nu_j L^j$$

and $\rho = \min\{\rho_1, \rho_2\}$.

Proposition 4 *The constraint (or cutoff) $\zeta^{\pi,\mu,\nu} \leq \rho$ can be added to the subproblem. If the subproblem is then infeasible, column generation terminates. If in addition $\rho = \rho_2$, node u can be pruned.*

Proof. From Observation 4, the lower bound for node u cannot be improved if $\lceil LB^{\pi,\mu,\nu} \rceil \geq z^u_{LP}(\tilde{Q})$. This is equivalent to the condition $LB^{\pi,\mu,\nu} > \lceil z^u_{LP}(\tilde{Q}) \rceil - 1$ or written explicitly:

$$\sum_{i \in M} \pi_i b_i + \sum_{j \in G^u_0} \mu_j K^j + \sum_{j \in H^u_0} \nu_j L^j + \min\{K^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0), L^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0)\}$$

$$> \lceil z^u_{LP}(\tilde{Q}) \rceil - 1$$

or $K^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0) > \delta_1$ and $L^0(\zeta^{\pi,\mu,\nu} + \mu_0 + \nu_0) > \delta_1$.
This in turn can be written as $\zeta^{\pi,\mu,\nu} > \frac{\delta_1}{K^0} - \mu_0 - \nu_0$ and $\zeta^{\pi,\mu,\nu} > \frac{\delta_1}{L^0} - \mu_0 - \nu_0$. Thus no improvement is obtained if $\zeta^{\pi,\mu,\nu} > \max\{\frac{\delta_1}{K^0}, \frac{\delta_1}{L^0}\} - \mu_0 - \nu_0 = \rho_1$, and thus the constraint $\zeta^{\pi,\mu,\nu} \leq \rho_1$ can be added. The other case is identical using Observation 5 in place of Observation 4. ∎

Note that when z is not restricted to be integer, the lower bound $LB^{\pi,\mu,\nu}$ can still be used in a similar way.

The basic algorithm can be modified based on the results of this section. Observation 4 can be used after solving the restricted Master LP to possibly terminate column generation, and Observation 5 to possibly prune the node. Then using the latest values of (π, μ, ν), ρ can be calculated, and used to introduce a cutoff constraint on the subproblem value. If the subproblem is infeasible, then again column generation terminates, and in addition the node is pruned if $\rho = \rho_2$. The implicit bound used at each node is $\lceil z^u_{LP}(Q) \rceil$.

4 The 0-1 Case

Various aspects of the algorithm can be simplified when $Q \subseteq \{0,1\}^m$, i.e. all the columns are 0-1 vectors corresponding to some subset of $M = \{1, \ldots, m\}$. Let $\overline{b} = \max_i b_i$.

First we examine the branching rule.

Proposition 5 *The branching vector $q^* \in \{0,1\}^m$ can be chosen so that $\sum_{i=1}^m q^*_i \leq \overline{b} + 1$.*

Proof. Let λ^* be the current fractional solution. Choose a minimal vector q^* with $\sum_{q:q \geq q^*} \lambda^*_q = \alpha \notin Z^1$ and let $S^* = \{i : q^*_i = 1\}$. WLOG suppose that $S^* = \{1, \ldots, r\}$. Consider the partial sums

$$\sum_{q:q_i=1,\ i \in \{1,\ldots,s\}} \lambda^*_q = f_s \quad s = 1, \ldots, r.$$

By hypothesis $f_s \in Z^1$ for $s = 1, \ldots, r-1$, as q^* is minimal.

If the values of f_s are distinct, we have $f_s \geq f_{s+1} + 1$ for $s = 1, \ldots, r-2$ and $f_{r-1} > f_r = \alpha > 0$. Thus $f_1 \geq r - 1$. As $\overline{b} \geq b_1 = f_1$, we obtain $r \leq \overline{b} + 1$.

Otherwise $f_s = f_{s-1}$ for some $s = 2, \ldots, r$. Then

$$\sum_{q:q_i=1,\ i\in\{1,\ldots,s\}} \lambda_q^* = \sum_{q:q_i=1,\ i\in\{1,\ldots,s-1\}} \lambda_q^*.$$

But then

$$\sum_{q:q_i=1,\ i\in S^*} \lambda_q^* = \sum_{q:q_i=1,\ i\in S^*\setminus\{s\}} \lambda_q^* = \alpha,$$

and q^* is not minimal, a contradiction. ∎

With $q \in \{0,1\}^m$, the additional constraints needed in the subproblem also simplify, and the auxiliary variables η^i needed in Proposition 2 can be eliminated.

Observation 6. $X_p = \{(x,z) : x \in \{0,1\}^m, z \in \{0,1\}, z \leq x_i \text{ if } p_i = 1, z \geq 1 - \sum_{i:p_i=1}(1-x_i)\}$.

Observations 1 and 3 also simplify.

Observation 7. If $K^j = 0$, then one can set $\lambda_q = 0$ for all $q \in Q(q^j)$ in (P^u), and drop z^j in the subproblem. The resulting set X_{q^j} reduces to x satisfying:

$$\{x \in \{0,1\}^m : \sum_{i:q_i^j=1} x_i \leq \sum_{i=1}^m q_i^j - 1\}$$

.

Observation 8. If $L^j = \min_{i:q_i^j=1} b_i$ for some $j \in H^u$, let $I = \{i : q_i^j = 1 \text{ and } b_i = L^j\} \neq \phi$. Then one can set $\lambda_q = 0$ for all $q \in Q \setminus Q(q^j)$ satisfying $\sum_{i\in I} q_i > 0$, and auxiliary variables are no longer needed in the subproblem. X_{q^j} reduces to x satisfying: $\{x \in \{0,1\}^m : x_i \geq x_k \text{ for all } i,k \text{ with } q_i^j = q_k^j = 1 \text{ and } k \in I\}$.

Specializing further, now suppose that $\overline{b} = 1$ i.e. $b_i = 1$ for all $i \in M$. Combining Proposition 5 with Observations 7 and 8 leads to one of the important branching schemes.

Proposition 6 *(Ryan and Foster [13]) When $\overline{b} = 1$, if λ is a fractional solution of $LP^u(Q)$, then there exists a pair of rows $\{u,v\} \subset M$ such that if q^* is defined by $q_u^* = q_v^* = 1$ and $q_i^* = 0$ otherwise*

i) $0 < \sum_{q \in Q(q^*)} \lambda_q < 1$
ii) in the separation step, on one branch $\sum_{q \in Q(q^*)} \lambda_q = 0$ *can be represented implicitly by removing all columns with* $q_u = q_v = 1$*, and adding the subproblem constraint* $x_u + x_v \leq 1$
iii) on the other branch $\sum_{q \in Q(q^*)} \lambda_q = 1$ *can be represented implicitly by removing all columns with* $q_u + q_v = 1$*, and adding the subproblem constraint* $x_u = x_v$.

5 Implementation

The algorithm and branching rule presented in Section 2 have been implemented as described with the branching pairs $(q^j, K^j), (q^j, L^j)$ chosen so that the support of q^j is as small as possible. Results for a small sample of cutting stock instances are presented in Table 1. The columns specify the name of the instance with the number of different items and the number of different types of rolls, then the number of nodes processed in the Branch and Bound algorithm, the number of Master iterations, the total number of generated columns, the number of times the subproblem is solved, the LP and IP values, the total CPU time (on a HP9000/712/80 using CPLEX 3.0), and finally the % of the total time spent in solving subproblems.

Inst	*BBnod*	*Mits*	*cols*	*SPs*	*LP*	*IP*	*Time*	*%SP*
4*b*1	5	14	10	12	6.74	7	1.0*s*	73.1
10*b*1	8	26	18	21	13.75	14	7*s*	92.4
14*b*1	27	70	43	47	50.63	51	55*s*	97.5
15*b*1	24	77	54	57	25.35	26	2*m*6*s*	98.9
20*b*1	20	72	50	55	115.05	116	1*m*39*s*	98.6
3*b*3	5	8	13	9	0	0	0.3*s*	33.3
4*b*7*c*	1417	1741	546	7798	702055	1674031	2*h*6*m*32*s*	98.6
4*b*7	401	1065	474	3780	1054	1462	48*m*52*s*	98.8
4*b*7*d*	7	31	64	119	81843	*infeas.*	19*s*	97.4
4*b*7*e*	77	120	156	735	188890	433274	8*m*44*s*	99.5

The first five instances in Table 1 are standard one-dimensional cutting stock problems, while the last five are instances of a model due to Hurkens[8], in which each roll is different and the cost of a cutting pattern is the size of the unused strip.

As one would expect, the last column shows that nearly all the computation time is spent in solving the mixed integer subproblems with a standard optimiser. We note that no instance has required using a separation with $|P^+| > 1$. Instance 4b7c has been solved by Hurkens using MINTO and a special purpose algorithm for the subproblem. His algorithm runs about about 10 times faster on this instance, but requires about twice as many nodes in the enumeration tree.

The early termination rules of Section 3 have also been implemented. Representative results from [17] for five instances of a telecommunications (edge partitioning) model described in Sutter et al.[15] are presented in Table 2. The first column gives the name of the instance including the number of nodes and edges in the network, the second the basic running time (CPU time in seconds on a Sparc 10 model 51, using CPLEX 2.1), the third the running time using the lower bound, and finally the time with both lower bound and subproblem cutoff.

Instance	*NoLB*	*LB*	*LB + Cutoff*
*ND*7*b*21	2.8	2.7	2.2
*ND*8*b*28	51.0	42.1	36.1
*ND*9*b*36	1056.1	701.4	615.2
*ND*12*b*50	41.0	39.5	34.6
*ND*15*b*72	1069.4	140.7	124.8

More extensive extensive computational results on different problems, branching rules, early termination, etc., can be found in Sutter et al.[15], Vanderbeck[17] and Vanderbeck and Wolsey[18].

6 Further Remarks

We emphasize that the algorithm presented in Section 2 generates an optimal integer solution for problem (P) even when the columns of Q are general integer, and not just 0-1, vectors. This means that the column generation scheme may have to generate points $x \in Q = \text{proj}_x(W)$ that are not extreme points of conv(Q), but are optimal subproblem solutions. This apparent contradiction is resolved once auxiliary variables w are permitted, and additional constraints or variables are added in the subproblem at node u of the enumeration tree.

An alternative and natural interpretation of the branching scheme proposed above is to introduce auxiliary integer variables $y(q^j) = \sum_{q \in Q(q^j)} \lambda_q$, and then branch on the variables $y(q^j)$. Such variables often arise naturally and may be a subset of the variables w appearing in the feasible region W of the subproblem: for instance in graph partitioning (Johnson et al.[9]) with $\overline{b} = 1$, M is the set of nodes, and the choice of a pair $\{u, v\}$ of nodes can be interpreted as branching on edge variables y_{uv} where $y_{uv} = 1$ if both nodes u and v lie in the same set of the partition, and $y_{uv} = 0$ if (u, v) is an edge of the corresponding multicut.

A second possibility is to combine the above algorithm with the polyhedral approach by adding cutting planes to the Master IP problem (P). Again taking the graph partitioning problem as an example, let $\chi_{uv} = 1 - y_{uv}$ represent the incidence vector of the multicut. Several families of valid inequalities $\sum_{e \in E} \pi_{uv} \chi_{uv} \geq \pi_0$ for the multicut polytope are known, so cuts $\sum_{e \in E} \pi_{uv}(1 - y_{uv}) \geq \pi_0$ can be added to (P), and the subproblems modified appropriately.

Various versions of the algorithm described here have been implemented. Several applications concerning the installation of multiplexing equipment are reported in Sutter at al. [15] involving both $\overline{b} = 1$ and general b. Results on implementation choices, heuristics and computational experience on different models including graph partitioning and multi-item lot-sizing are reported in Vanderbeck [17]. Vanderbeck and Wolsey [18] will include examples of implementation of the proposed branching scheme and practical comparison of branching rules.

Acknowledgement. We are grateful to C. Hurkens, M. Savelsbergh, and a referee for their helpful comments and criticism, and to C. Hurkens for providing information and data for the generalised cutting stock model.

7 References

[1] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, Branch and Price: Column Generation for Solving Huge Integer Programs, Computational Optimization Center COC-94-03, Georgia Institute of Technology, Atlanta, February 1994 (revised May 1995).

[2] G.B. Dantzig and P. Wolfe, Decomposition Principle for Linear Programs, Operations Research 8, 101-111 (1960).

[3] J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis, Time Constrained Routing and Scheduling, Chapter 11 in *Handbooks in Operations Research and Managament Science: Networks* eds. M.E. Ball, T.L. Magnanti, C. Monma and G.L. Nemhauser, North-Holland (1994).

[4] A.A. Farley, A Note on Bounding a Class of Linear Programming Problems, including Cutting Stock Problems, Operations Research 38, 922-923 (1990).

[5] P.C. Gilmore and R.E. Gomory, A Linear Programming Approach to the Cutting Stock Problem, Operations Research 9, 849-859 (1961).

[6] P.C. Gilmore and R.E. Gomory, A Linear Programming Approach to the Cutting Stock Problem: Part II, Operations Research 11, 863-888 (1963).

[7] P. Hansen, B. Jaumard and M. Poggi de Aragao, Mixed Integer Column Generation and the Probabilistic Maximum Satisfiability Problem, Proceedings of $IPCO2$, Carnegie-Mellon University, Pittsburg, 165-180 (1992).

[8] Hurkens, private communication (1995).

[9] E. Johnson, A. Mehrotra and G.L. Nemhauser, Min-cut Clustering, Mathematical Programming 62, 133-152.

[10] L.S. Lasdon, Optimization Theory for Large Systems, Macmillan, London 1970.

[11] T.L. Magnanti and L.A. Wolsey, Optimal Trees, Chapter 9 in *Network Models, Handbooks in Operations Research and Managament Science 7* M. Ball et al. (eds.), North-Holland, Amsterdam (1995).

[12] M. Minoux, A Class of Combinatorial Optimization Problems with Polynomially Solvable Large Scale Set-Covering/Partitioning Relaxations,

RAIRO 21, 105-136 (1987).

[13] D.M. Ryan and B.A. Foster, An Integer Programming Approach to Scheduling, A. Wren (ed.) *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, North-Holland, Amsterdam, 269-280 (1981).

[14] M.W.P. Savelsbergh, A Branch and Price Algorithm for the Generalized Assignment Problem, Computational Optimization Center COC-93-02, Georgia Institute of Technology, Atlanta (1993).

[15] A. Sutter, F. Vanderbeck and L.A. Wolsey, Optimal Placement of Add/Drop Multiplexers: Heuristic and Exact Algorithms, Core Discussion Paper 9479, Université Catholique de Louvain, Louvain-la-Neuve 1994.

[16] P.H. Vance, C. Barnhart, E.L. Johnson and G.L. Nemhauser, Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound, Computational Optimization Center COC-92-09, Georgia Institute of Technology, Atlanta (1992).

[17] F. Vanderbeck, Decomposition and Column Generation for Integer Programs, Ph.D. Thesis, Faculté des Sciences Appliqées, Université Catholique de Louvain, Louvain-la-Neuve (1994).

[18] F. Vanderbeck and L.A. Wolsey, ???