Column Generation in Integer Programming with Applications in Multicriteria Optimization

Matthias Ehrgott
Department of Engineering Science
The University of Auckland, New Zealand
email: m.ehrgott@auckland.ac.nz
and
Laboratoire d'Informatique de Nantes Atlantique
Université de Nantes, France
email: matthias.ehrgott@univ-nantes.fr

Jørgen Tind
Department of Mathematical Sciences
University of Copenhagen, Denmark
email: tind@math.ku.dk

March 26, 2007

Abstract

This paper presents in a unified form a column generation scheme for integer programming. The scheme incorporates the two major algorithmic approaches in integer programming, the branch and bound technique and the cutting plane technique. With integrality conditions imposed on the variables it is of importance to limit the number of columns introduced in the integer programming problem. This is equally important in the case of multiple criteria where usually multiple alternative efficient solutions are required. The suggested scheme gives additional dual information that limits the work required to move among the alternatives to be generated.

Keywords: Column generation, integer programming, multicriteria optimization, branch and bound, cutting planes, saddle points.

MSC 2000: 90C29, 90C10

1   Introduction

This paper presents a column generation approach for integer programming.

The paper consists of two parts. The first part introduces the basic principles in the single criterion case. It is based on the application of the two standard techniques in integer programming, branch and bound as well as cutting planes. A separate description is given for each case, respectively. The branch and bound approach falls within the class of branch and price algorithms. See for example Barnhart *et al.* (1998), who give an exposition of the area. They also discuss various branching schemes and related enumeration problems to avoid possible regeneration of columns that have already been cancelled higher up in the branch and bound tree. In the current setting we keep track of the branchings by keeping this information as separate constraints in the subproblems. Vanderbeck and Wolsey (1996) also consider branching schemes. Vanderbeck (2005) summarizes much of the most recent work. Alternative formulations are developed by Villeneuve *et al.* (2005).

Equally, one may say that the cutting plane approach is within the class of cut and price algorithms. Less previous work exists for this kind of procedures, but the work by Ralphs and Galati (2006) on a dynamic procedure should be mentioned.

In a separate section we also emphasize the relationship with the classical column generation and decomposition methods in linear programming and convex programming by demonstrating a necessary and sufficient saddle point condition for optimality of the procedure.

The second part of the paper points out the potential use of these techniques in multiobjective optimization with integer constraints. As is well known a weighted sums approach may exclude some efficient points from consideration. More general methods must be applied, for example the ε-constrained procedure or the Tchebycheff procedure, see Ehrgott (2005). For general multicriteria optimization with integer constraints it is of particular importance to limit the number of integer variables. This is the core of column generation to generate the appropriate set of columns which may otherwise not be detected using the standard LP relaxation of the master program.

Once an efficient solution has been found together with the appropriate dual information we demonstrate by sensitivity analysis how neighboring efficient solutions may be obtained without going back to a change of original parameters. This is specifically demonstrated for the ε-constraint approach.

2   Column Generation in Branch and Bound

Let $c_j \in \mathbb{R}$ and $a^j \in \mathbb{R}^m$ for $j = 1 \ldots n$ and $b \in \mathbb{R}^m$ be constants and let $x = (x_1 \ldots x_n)$ denote a set of variables. Consider an integer programming problem of the form

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a^j x_j \geq b \tag{1}$$

$$x_j \geq 0, \text{ integer.}$$

Just for simplicity assume that the feasible set $X = \{x \in \mathbb{R}^n : \sum_{j=1}^{n} a^j x_j \geq b, x_j \geq 0, \text{ integer}\}$ is nonempty and bounded. Hence an optimal solution exists. We are going to treat this problem by a standard branch and bound technique based on linear programming and separation of the original variables. Then at any stage a finite set exists of terminals or end nodes of a branch and bound tree. Let T denote this set. For each $t \in T$ there exists a corresponding linear programming subproblem (P_t):

$$\min_{x} \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a^j x_j \geq b$$

$$l_j^t \leq x_j \leq u_j^t$$

$$x \geq 0$$

where $l_j^t \in \mathbb{R}$ and $u_j^t \in \mathbb{R}$ are the separating restrictions on the variables at the corresponding node. For ease of exposition we keep the sign restriction on the x variables separately and do not include them in the separation constraint. Let X_t denote the set of feasible solutions of (P_t). No integer solution of (1) is omitted by separation which means that $X \subset \cup_{t \in T} X_t$. Let $\mathcal{T}$ denote the collection of possible sets T of subproblems. The branch and bound process terminates if all problems (P_t) are infeasible, or if some (P_t) has an optimal solution that is integer, and has an objective function value which is the lowest among the values of all feasible suproblems. So, the

solution of IP (1) by branch and bound can be understood as finding $T \in \mathcal{T}$ such that the smallest optimal value of any (P_t), $t \in T$ is largest, i.e.,

$$\max_{T \in \mathcal{T}} \min_{t \in T} \min_{x} \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a^j x_j \;\geq\; b \tag{2}$$

$$l_j^t \;\leq\; x_j \;\leq\; u_j^t$$

$$x \;\geq\; 0.$$

By our assumptions two cases occur.

Case 1: (P_t) has a finite optimal value. Consider the dual

$$\max_{y^t, w^t, v^t} by^t + \sum_{j=1}^{n} w_j^t l_j^t - \sum v_j^t u_j^t$$

$$\text{s.t.} \; a^j y^t + w_j^t - v_j^t \;\leq\; c_j \text{ for } j = 1 \ldots n$$

$$y^t, w^t, v^t \;\geq\; 0$$

with variables $y^t \in \mathbb{R}^m, w_j^t \in \mathbb{R}$ and $v_j^t \in \mathbb{R}$ for $j = 1 \ldots n$ and $t = 1 \ldots |T|$.

Let $(\bar{y}^t \geq 0, \bar{w}_j^t \geq 0, \bar{v}_j^t \geq 0)$ be an optimal solution of the dual. Note that $\bar{w}_j^t \cdot \bar{v}_j^t = 0$.

Case 2: (P_t) is infeasible. Then, with the same notation using Farkas' Lemma there exist $\bar{y}^t, \bar{w}^t, \bar{v}^t \geq 0$ such that

$$a^j \bar{y}^t + \bar{w}_j^t - \bar{v}_j^t \;\leq\; 0 \text{ for } j = 1, \ldots, n$$

$$b\bar{y}^t + \sum_{j=1}^{n} \bar{w}_j^t l_j^t - \sum_{j=1}^{n} \bar{v}_j^t u_j^t \;>\; 0.$$

Combining cases 1 and 2 we can reformulate (2) as

$$\max_{T \in \mathcal{T}} \min_{t \in T} \max_{y^t, w^t, v^t} by^t + \sum_{j=1}^{n} w_j^t l_j^t - \sum_{j=1}^{n} v_j^t u_j^t$$

$$\text{s.t.} \; a^j y^t + w_j^t - v_j^t \;\leq\; c_j' \text{ for } j =, \ldots, n$$

where $c'_j = c_j$ if (P_t) has an optimal solution and $c'_j = 0$ if (P_t) is infeasible.

The dual variables w^t_j and v^t_j are specific for column j at terminal t. When a new column is to be considered for introduction without any explicit prior bounds on the corresponding variables we leave those variables out of consideration. So, we are left with the question: Does there exist a, c such that $a\bar{y}^t > c'$. For each (P_t) this is done by solving the problem

$$\max_{a,c} a\bar{y}^t - c',$$

where the maximization is over all "legal" columns (a, c) of the IP (1). In view of formulation (2) we can generate a column to be added in every (P_t). In total it is enough to solve

$$\max_{t \in T} \max_{a,c} a\bar{y}^t - c'.$$

If the optimal value is less than or equal to 0, then there is no column to be generated. If we now assume that the set A of columns of the IP can be described by some polyhedron, $A = \{a \in \mathbb{R}^m : Ba \leq d\}$ we can write the column generation problem as

$$\max_{t \in T} \max_{a} a\bar{y}^t - c'$$
$$\text{s.t. } Ba \leq d, \tag{3}$$

where $c' = \begin{cases} c & (P_t) \text{ feasible} \\ 0 & (P_t) \text{ infeasible}. \end{cases}$

In some applications $c = 1$ and independant of the column a. In other cases c is a linear function of the column a. Consider the dual of the inner optimization of (3). We then get the form

$$\max_{t \in T} \left(\min_{u^t \geq 0} du^t - c' \right)$$
$$\text{s.t. } u^t B = \bar{y}^t, \ t \in T.$$

If preferable the problem can be reformulated as a linear programming problem

$$\max_{u^t \geq 0, s \in \mathbb{R}} s$$
$$\text{s.t. } s \leq du^t - c', \ t \in T$$
$$u^t B = \bar{y}^t, \ t \in T.$$

Example 1 *We consider a cutting stock problem of cutting items of length 6, 9 and 12 out of pieces of length 20. 6, 3 and 2 items are required, respectively. There are four possible cutting patterns and the IP is as follows.*

$$
\begin{array}{rrcrcrcrcl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{s.t.} & 3x_1 & + & x_2 & + & x_3 & & & \geq & 6 \\
& & & x_2 & & & + & 2x_4 & \geq & 3 \\
& & & & & x_3 & & & \geq & 2 \\
& & & & & & & x_j & \geq & 0 \ and \ integer.
\end{array}
\tag{4}
$$

The optimal IP solution is $x_1 = x_2 = x_4 = 1$ and $x_3 = 2$ with objective $z = 5$. The LP relaxation has optimal solution $x_1 = 1\frac{1}{3}, x_3 = 2, x_4 = 1.5$ with $z = 4\frac{5}{6}$.

Now assume that we want to solve the problem by LP-based branch and bound and column generation and that initially we only have x_1, x_3, x_4 in the problem. The LP relaxation has the same optimal solution as the original problem. Optimal dual values are $\bar{y}_1 = \frac{1}{3}, \bar{y}_2 = 0.5, \bar{y}_3 = \frac{2}{3}$. Since $\bar{y}_1 + \bar{y}_2 < 1$ the column of x_2 cannot be generated. We shall now continue by branching on variable x_4.

Branch by $x_4 \geq 2$:

$$
\begin{array}{rrcrcrcl}
\min & x_1 & + & x_3 & + & x_4 & & \\
\text{s.t.} & 3x_1 & + & x_3 & & & \geq & 6 \\
& & & & & 2x_4 & \geq & 3 \\
& & & x_3 & & & \geq & 2 \\
& & & & & x_4 & \geq & 2 \\
& & & & & x_j & \geq & 0.
\end{array}
$$

The optimal solution is $(x_1, x_3, x_4) = (1\frac{1}{3}, 2, 2)$, $(\bar{y}_1, \bar{y}_2, \bar{y}_3) = (\frac{1}{3}, 0, \frac{2}{3})$ and $\bar{w}_4 = 1$. The value is equal to $5\frac{1}{3}$.

We shall now check if column 2 is eligible for insertion:

$\sum_{i=1}^{3} \bar{y}_i a_{i2} = \frac{1}{3} \cdot 1 + 0 \cdot 1 + \frac{2}{3} \cdot 0 = \frac{1}{3} \leq 1 = c_2$. Hence column 2 is not introduced in this problem.

6

Branch by $x_4 \leq 1$:

$$
\begin{array}{rrrrrcr}
\min & x_1 & + & x_3 & + & x_4 & & \\
\text{s.t.} & 3x_1 & + & x_3 & & & \geq & 6 \\
& & & & & 2x_4 & \geq & 3 \\
& & & x_3 & & & \geq & 2 \\
& & & & & x_4 & \leq & 1 \\
& & & & & x_1 & \geq & 0.
\end{array}
$$

No feasible solution exists.

By Farkas' lemma the following system of inequalities has a solution.

$$
\begin{array}{rcl}
3y_1 & \leq & 0 \\
y_1 + y_3 & \leq & 0 \\
2y_2 - v_4 & \leq & 0 \\
6y_1 + 3y_2 + 2y_3 - v_4 & > & 0,
\end{array}
$$

where v_4 is the dual of $x_4 \leq 1$. A possible solution is $(\bar{y}_1, \bar{y}_2, \bar{y}_3) = (0, \frac{1}{2}, 0)$ and $\bar{v}_4 = 1$.

We shall check column 2 by calculating $\sum_{i=1}^{3} \bar{y}_i a_{i2} = 0 \cdot 1 + \frac{1}{2} \cdot 1 + 0 \cdot 0 = 0.5 > 0$. Hence column 2 is introduced and we obtain the problem

$$
\begin{array}{rrrrrrrrrcr}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{s.t.} & 3x_1 & + & x_2 & + & x_3 & & & \geq & 6 \\
& & & x_2 & + & & + & 2x_4 & \geq & 3 \\
& & & & & x_3 & & & \geq & 2 \\
& & & & & & & x_4 & \leq & 1 \\
& & & & & & & x_j & \geq & 0.
\end{array}
$$

An optimal and integer solution is $(x_1, x_2, x_3, x_4) = (1, 1, 2, 1)$ with objective value $= 5$. Since this is the lowest of the values of the two terminals this solution is proven to be optimal for the original problem (4).

3 Column Generation and Cutting Planes

Consider again the original integer programming problem (1).

In this section we are going to solve (1) by means of a cutting plane technique. In the column generation approach we begin by solving the integer programming problem (1) with a subset of columns and continue to introduce additional columns until a termination criterion proves optimality of the entire problem (1). The main goal here is to outline the technique leading to termination and proof of optimality without explicit introduction of all columns; here by using cutting plane methods.

As it is well known, the idea behind cutting plane techniques is to introduce valid inequalities which remove some nonintegral parts of the feasible set in the linear programming relaxation of (1). There are many ways in which this can be done. We are going to use one of the classical approaches by using Chvatal-Gomory cuts, C-G cuts, see Nemhauser and Wolsey (1988).

The valid inequalities have their coefficients determined according to some specific rules. These rules imply that no integer solution is removed by introduction of an inequality. The jth coefficient in the inequality is defined by insertion of the elements from the corresponding original column a^j into a specific function, which is generated recursively through the following operations.

- Multiplication by a non-negative scalar,

- addition,

- application of the round up operation to nearest integer above $\lceil \cdot \rceil$. For example $\lceil 7.6 \rceil = 8$.

Example 2 *For a problem with $m = 2$ we may consider the variables $d = (d_1, d_2) \in \mathbb{R}^2$. A C-G function F in d is*

$$F(d_1, d_2) = \lceil 2\lceil d_1 \rceil + 3.5\lceil d_2 \rceil \rceil.$$

With $(d_1, d_2) = (0.5, 0.7)$ we get

$$F(0.5, 0.7) = \lceil 2\lceil 0.5 \rceil + 3.5\lceil 0.7 \rceil \rceil = \lceil 2 \times 1 + 3.5 \times 1 \rceil = \lceil 5.5 \rceil = 6.$$

Let $\mathcal{F}$ denote the set of all C-G functions and consider a function $F \in \mathcal{F}$. F then defines the following valid inequality.

$$\sum_{j=1}^{n} F(a^j)x_j \geq F(b).$$

By its construction this inequality is satisfied by all integer feasible solutions of (1).

The aim is to find a subset $\tilde{\mathcal{F}} \subseteq \mathcal{F}$ of functions such that the following linear programming problem has an optimal solution, which is integer and hence also optimal for the original problem (1).

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a^j x_j \geq b \\
& \sum_{j=1}^{n} F(a^j) x_j \geq F(b) \text{ for all } F \in \tilde{\mathcal{F}} \\
& x_j \geq 0 \text{ for } j = 1, \ldots, n.
\end{aligned}
\tag{5}
$$

In analogy with (2) we then have that (1) can be converted into the problem

$$
\begin{aligned}
\max_{\tilde{\mathcal{F}} \subseteq \mathcal{F}} \min_{x \geq 0} \quad & \sum_{j=1}^{n} c_j x_j \\
& \sum_{j=1}^{n} a^j x_j \geq b \\
& \sum_{j=1}^{n} F(a^j) x_j \geq F(b) \text{ for all } F \in \tilde{\mathcal{F}}.
\end{aligned}
\tag{6}
$$

For a given set $\tilde{\mathcal{F}}$ the set of feasible solutions of (6) includes the convex hull of feasible solutions of (1). With rational data the theory of C-G cutting planes says that a finite set $\tilde{\mathcal{F}}$ exists such that the feasible solution set of (6) is equal to the convex hull of the feasible solutions in (1). Hence it is sufficient for a given problem (1) to consider only *finite* sets $\tilde{\mathcal{F}}$ in order to solve (6). Furthermore, the inner problem of (6) may be dualized by linear programming duality so that (6) is converted into the form

$$
\begin{aligned}
\max_{\tilde{\mathcal{F}} \subseteq \mathcal{F}} \max_{u_F \geq 0, v \geq 0} \quad & \sum_{F \in \tilde{\mathcal{F}}} u_F F(b) + vb \\
\text{s.t.} \quad & \sum_{F \in \tilde{\mathcal{F}}} u_F F(a^j) + va^j \leq c_j \text{ for } j = 1, \ldots, n.
\end{aligned}
\tag{7}
$$

Define the function $G : \mathbb{R}^m \to \mathbb{R}$ as

$$G(d) = \sum_{F \in \tilde{\mathcal{F}}} u_F F(d) + vd$$

and note that $G(d)$ itself is a C-G function. Hence (7) can be changed into the form

$$\begin{aligned}
\max_{G \in \mathcal{F}} \quad & G(b) \\
\text{s.t.} \quad & G(a^j) \le c_j \text{ for } j = 1, \ldots, n.
\end{aligned} \tag{8}$$

This is the "classical" dual form of an integer programming problem, see for example Schrijver (1986).

In a column generation framework (8) is solved with a selection of columns a^j. If in addition the constraints of (8) are satisfied for all columns a^j then the optimal value of (1) has been obtained, and an optimal solution x of (1) can be found by solving the corresponding linear program (5). However, if a column a^j exists such that $G(a^j) > c_j$ this column should be introduced.

Example 3 *Consider the cutting stock example again.*

$$\begin{array}{rrrrrrrrl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & \\
\text{s.t.} & 3x_1 & + & x_2 & + & x_3 & & & \ge \quad 6 \\
& & & x_2 & & & + & 2x_4 & \ge \quad 3 \\
& & & & & x_3 & & & \ge \quad 2 \\
& & & & & & & x_j & \ge \quad 0 \text{ for } j = 1, \ldots, 4.
\end{array}$$

With the column a^2 corresponding to x_2 removed we get the reduced problem

$$\begin{array}{rrrrrrl}
\min & x_1 & + & x_3 & + & x_4 & \\
\text{s.t.} & 3x_1 & + & x_3 & & & \ge \quad 6 \\
& & & & & 2x_4 & \ge \quad 3 \\
& & & x_3 & & & \ge \quad 2 \\
& & & & & x_j & \ge \quad 0 \text{ for } j = 1, 3, 4.
\end{array}$$

An optimal solution of this last problem is again $(x_1, x_3, x_4) = (1\frac{1}{3}, 2, 2)$ with an objective function value $5\frac{1}{3}$.

Here $m = 3$ and by application of the C-G function $F_1(d_1, d_2, d_3) = \lceil \frac{1}{3}d_1 + \frac{2}{3}d_3 \rceil$ we generate the following valid inequality

$$\left\lceil \frac{1}{3} \times 3 \right\rceil x_1 + \left\lceil \frac{1}{3} \times 1 + \frac{2}{3} \times 1 \right\rceil x_3 \ge \left\lceil \frac{1}{3} \times 6 + \frac{2}{3} \times 2 \right\rceil$$

or
$$x_1 + x_3 \geq 4.$$

With the C-G function $F_2(d_1, d_2, d_3) = \lceil \frac{1}{2} d_2 \rceil$ we get the additional valid inequality

$$\left\lceil \frac{1}{2} \times 2 \right\rceil x_4 \geq \left\lceil \frac{1}{2} \times 3 \right\rceil$$

or

$$x_4 \geq 2.$$

By adding those two new inequalities into the constraints we obtain

$$
\begin{array}{rcrcrcrcl}
\min & x_1 & + & x_3 & + & x_4 & & \\
\text{s.t.} & 3x_1 & + & x_3 & & & \geq & 6 \\
& & & & & 2x_4 & \geq & 3 \\
& & & x_3 & & & \geq & 2 \\
& x_1 & + & x_3 & & & \geq & 4 \\
& & & & & x_4 & \geq & 2 \\
& & & & & x_j & \geq & 0 \ \textit{for } j = 1, 3, 4.
\end{array}
\tag{9}
$$

By solving this linear programming problem we get the optimal and also integer solution $x_1 = x_3 = x_4 = 2$ and objective value equal to 6. The dual variables are

$$(y_1, y_2, y_3, u_{F_1}, u_{F_2}) = (0, 0, 0, 1, 1)$$

Hence the appropriate C-G function is defined by

$$G(d) = yd + u_{F_1} F_1(d) + u_{F_2} F_2(d) = F_1(d) + F_2(d) = \left\lceil \frac{1}{3} d_1 + \frac{2}{3} d_3 \right\rceil + \left\lceil \frac{1}{2} d_2 \right\rceil.$$

With $a^2 = (1, 1, 0)$ we have $G(1, 1, 0) = \lceil \frac{1}{3} \times 1 + \frac{2}{3} \times 0 \rceil + \lceil \frac{1}{2} \times 1 \rceil = 1 + 1 = 2$. Since $c_2 = 1$ we get that $G(a_2) > c_2$ violating the constraints of (8). Hence column a^2 should be introduced in (9). In order to calculate the coefficients of the cuts we calculate

$$F_1(1, 1, 0) = \left\lceil \frac{1}{3} \times 1 + \frac{2}{3} \times 0 \right\rceil = 1$$

and

$$F_2(1, 1, 0) = \left\lceil \frac{1}{2} \times 1 \right\rceil = 1$$

11

and we get the problem

$$
\begin{array}{rrrrrrrrrl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
s.t. & 3x_1 & + & x_2 & + & x_3 & & & \geq & 6 \\
& & & x_2 & & & + & 2x_4 & \geq & 3 \\
& & & & & x_3 & & & \geq & 2 \\
& x_1 & + & x_2 & + & x_3 & & & \geq & 4 \\
& & & x_2 & & & + & x_4 & \geq & 2 \\
& & & & & & & x_j & \geq & 0 \; for \; j = 1, 2, 3, 4.
\end{array}
\tag{10}
$$

By solving (10) we get the optimal and also integer solution $(x_1, x_2, x_3, x_4) = (1, 1, 2, 1)$ with objective function value 5. The corresponding dual variables are

$$(y_1, y_2, y_3, u_{F_1}, u_{F_2}) = (0.25, 0.5, 0.5, 0.25, 0).$$

Hence we get the corresponding function

$$
\begin{aligned}
G(d_1, d_2, d_3) &= y_1 d_1 + y_2 d_2 + y_3 d_3 + u_{F_1} F_1(d_1, d_2, d_3) + u_{F_2} F_2(d_1, d_2, d_3) \\
&= 0.25 d_1 + 0.5 d_2 + 0.5 d_3 + 0.25 \left[\frac{1}{3} d_1 + \frac{2}{3} d_3 \right] + 0.
\end{aligned}
$$

Observe, for example, that $G(a^1) = G(3, 0, 0) = 0.25 \times 3 + 0.25 \left[\frac{1}{3} \times 3 + \frac{2}{3} \times 0 \right] = 1$. Similarly $G(a^2) = G(a^3) = G(a^4) = 1$, implying that all columns satisfy the constraints of (8). This proves optimality of the solution also for the original problem (4).

4   Saddlepoints and Column Generation

The purpose of decomposition in the current setup is via a master problem to find a tree for which dual feasiblity is checked by the subproblem.

In linear and convex programming the master problem finds a set of primal and dual variables of which the primal variables are feasible. Dual feasibility is then examined by the subproblem. Usually each master problem is solved to optimality. The primal-dual solutions then define a saddlepoint for the Langrangean function of the master problem. The purpose of the subproblem is to check whether it is still a valid saddlepoint when all solutions are taken into account.

This section establishes a similar setup for integer programming. This shall first be illustrated by application of the branch and bound technique,

and we shall see how a saddlepoint is generated. The end of this section treats the cutting plane case too.

Consider again the original problem in the form (2). Observe that it does not matter if all the inner problems share the same variables. This means that (2) can be reformulated to

$$z_1 = \max_{T \in \mathcal{T}} \min_{x} \min_{t \in T} \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t. } \sum_{j=1}^{n} a^j x_j \geq b \qquad (11)$$

$$l_j^t \leq x_j \leq u_j^t$$

$$x \geq 0.$$

If the inner problem is infeasible for a $t \in T$ we then set the objective function value of the inner problem to $+\infty$.

By an interchange of the first two operations we get the problem

$$z_2 = \min_{x} \max_{T \in \mathcal{T}} \min_{t \in T} \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t. } \sum_{j=1}^{n} a^j x_j \geq b \qquad (12)$$

$$l_j^t \leq x_j \leq u_j^t$$

$$x \geq 0.$$

This interchange implies in general that $z_1 \leq z_2$. Let (T^*, x^*) denote an optimal solution of (11) obtained by a tree with nodes T^* and resulting in the (primal) optimal solution x^*. For $x = x^*$ we get in (12) the same objective value for any valid tree. Hence, $z_1 = z_2$.

By Lagrangean duality (12) is equivalent with

$$\max_{T \in \mathcal{T}, y^t \geq 0, w^t \geq 0, v^t \geq 0} \left\{ \min_{x \geq 0} \min_{t \in T} by^t + \sum_{j=1}^{n} w_j^t l_j^t - \sum_{j=1} v_j^t u_j^t + \sum_{j=1}^{n} (c_j - y^t a_j - w_j^t + v_j^t) x_j \right\}.$$

$$(13)$$

13

Let $\mathcal{N} = \{(T, y^t, w^t, v^t) \mid y^t, w^t, v^t \geq 0 \text{ for } t = 1, \ldots, |T|, T \in \mathcal{T}\}$ and let $S \in \mathcal{N}$. Let also $K((T, y^t, w^t, v^t), x) = \min_{t \in T} by^t + \sum_{j=1}^{n} w_j^t l_j^t - \sum_{j=1}^{n} v_j^t u_j^t + \sum_{j=1}^{n} (c_j - y^t a_j - w_j^t + v_j^t) x_j$.

In this way the Lagrangean dual (13) undertakes the short form

$$\max_{S \in \mathcal{N}} \min_{x \geq 0} K(S, x). \tag{14}$$

The program (12) is directly equivalent with

$$\min_{x \geq 0} \quad \max_{T \in \mathcal{T}, y^t \geq 0, w^t \geq 0, v^t \geq 0}$$

$$\min_{t \in T} by^t + \sum_{j=1}^{n} w_j^t l_j^t - \sum_{j=1}^{n} v_j^t u_j^t + \sum_{j=1}^{n} (c_j - y^t a_j - w^t + v_j^t) x_j \tag{15}$$

or in short form

$$\min_{x \geq 0} \max_{S \in \mathcal{N}} K(S, x). \tag{16}$$

The observations above can be summarized into the following

Proposition 1 x^* *is an optimal solution of the original program* (1) *if and only if there exists a tree with nodes and dual variables* S^* *that are optimal in* (16) *and* (14).

In this situation we thus get that (16) and (14) have equal values and that

$$\min_{x \geq 0} \max_{S \in T} K(S, x) = \max_{S \in \mathcal{N}} \min_{x \geq 0} K(S, x) = K(S^*, x^*).$$

Alternatively this expresses that (S^*, x^*) is a saddlepoint for $K(S, x)$, i.e.

$$K(S, x^*) \leq K(S^*, x^*) \leq K(S^*, x) \text{ for all } S \in \mathcal{N} \text{ and } x \geq 0. \tag{17}$$

So, alternatively we have

Proposition 2 x^* *is an optimal solution of the original program* (1) *if and only if there exists a tree with nodes and dual variables* S^* *such that* (S^*, x^*) *is a saddlepoint for* $K(S, x)$.

In the column generation approach presented here we gradually increase the dimension of the variables x by the introduction of new columns. Let C

14

be the current restricted set. At optimality we get a saddlepoint with this set C, i.e. the master program obtains a solution (S^*, x^*) which satisfies

$$K(S, x^*) \leq K(S^*, x^*) \leq K(S^*, x) \text{ for all } S \in \mathcal{N} \text{ and } x \in C.$$

The column generation subproblem checks whether this is valid for all candidate columns and corresponding variables. If not, then the right inequality of (17) is violated and a column and a new variable is introduced in the function $K(S, x)$.

The solution process stops when no violation occurs of the right inequality of (17).

A similar process develops by using the cutting plane technique. We shall here consider a general Lagrange function L of the type

$$L(G, x) = \sum_{j=1}^{n} (c_j - G(a^j)) x_j + G(b)$$

where $G : \mathbb{R}^m \to \mathbb{R}$ is a C-G function as considered before. This is a generalization of the standard Lagrange function in which the ordinary Langrange multipliers have been replaced by a C-G function.

The Langrange dual is

$$\max_{G \in \mathcal{F}} \min_{x \geq 0} L(G, x).$$

The original problem (1) is directly equivalent with the primal form

$$\min_{x \geq 0} \max_{G \in \mathcal{F}} L(G, x).$$

For a given set of columns we have by duality that an optimal solution x^* exists if and only if there exists a C-G function G forming a saddlepoint, i.e.

$$L(G, x^*) \leq L(G^*, x^*) \leq L(G^*, x) \text{ for } x \geq 0 \text{ and } G \in \mathcal{F}.$$

If a new column exists with $G(a_j) > c_j$ this violates the right hand side of the saddlepoint and must therefore be included. The procedure stops when the saddlepoint is no longer violated by any column.

5 Application to Multiobjective Integer Programming

We use the scheme of the previous section in a multiobjective context. Consider a multiobjective IP

$$\min \sum_{j=1}^{n} c^j x_j$$

$$\text{s.t. } \sum_{j=1}^{n} a^j x_j \geq b \tag{18}$$

$$x_j \geq 0, \text{ integer,}$$

where $c^j \in \mathbb{R}^p$. A feasible solution x of (18) is called efficient if there is no other feasible solution x' such that $\sum_{j=1}^{n} c^j x'_j \leq \sum_{j=1}^{n} c^j x_j$ and at least one of these inequalities is strict. The set of efficient solutions is usually found by scalarization techniques. A well known technique is the ε-constraint method. All efficient solutions can be found using this method (Chankong and Haimes, 1983).

The scalarized problem can be written as follows.

$$\min \sum_{j=1}^{n} c_i^j x_j$$

$$\text{s.t. } \sum_{j=1}^{n} -\hat{c}^j x_j \geq -\varepsilon \tag{19}$$

$$\sum_{j=1}^{n} a^j x_j \geq b$$

$$x_j \geq 0, \text{ integer,}$$

where $\hat{c}^j = (c_1^j, \ldots, c_{i-1}^j, c_{i+1}^j, \ldots, c_p^j)$ and $\varepsilon \in \mathbb{R}^{p-1}$. It is well known that for every efficient solution x of (MOIP) there exist i and ε such that x is an optimal solution of (19), see Ehrgott (2005).

Solving (19) by branch and bound, we can apply the column generation framework of the previous section and obtain the column generation subproblem

$$\max_{t \in T} \max_{a,c} a\bar{y}^t - \hat{c}\bar{\omega}^t - c_i^{j'},$$

16

where $\bar{\omega}^t$ are the optimal dual values of the ε-constraints in node t of T and $c_i^{j'}$ is defined analogously to c_j' in Section 2.

Assuming that both a and $\hat{c}$ have some polyhedral description, i.e., $AC = \{(a, \hat{c}) \in \mathbb{R}^{m+p-1} : B(a, \hat{c}) \le d\}$ we obtain

$$\max_{t \in T} \max_{a,c} a\bar{y}^t - \hat{c}\bar{\omega}^t - c_i^{j'}$$
$$B(a, c) \le d$$

or, with the same arguments as before

$$\max s$$
$$\text{s.t. } s \le du^t - c_i^{j'} \text{ for } t \in T$$
$$u^t B = (\bar{y}^t, -\bar{\omega}^t) \text{ for } t \in T.$$

Solving (19) by cutting planes we can proceed as in Section 3 with C-G functions incorporating $\hat{c}_j$ and ε generating cutting planes

$$F(-\hat{c}^j, a^j) \ge F(-\varepsilon, b).$$

In the following two examples we illustrate both the cutting plane and branch and bound approach.

Example 4 *Consider the following bicriterion problem.*

$$
\begin{array}{rrrrrrrl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 \\
\min & 14x_1 & + & 4x_2 & & & + & 2x_4 \\
\text{s.t.} & 3x_1 & + & x_2 & + & x_3 & & \ge 6 \\
& & & x_2 & + & & + 2x_4 & \ge 3 \\
& & & & & x_3 & & \ge 2 \\
& & & & & & x_j & \ge 0 \text{ and integer.}
\end{array}
\tag{20}
$$

The problem has the following four efficient solutions

x_1	x_2	x_3	x_4	$c^1 x$	$c^2 x$
0	0	6	2	8	4
0	1	5	1	7	6
0	3	3	0	6	12
1	1	2	1	5	20.

17

We shall use the ϵ-constraint method specifying a maximum size 13 of the second objective.

$$
\begin{array}{rrrrrrrrrl}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{s.t.} & -14x_1 & - & 4x_2 & & & - & 2x_4 & \geq & -13 \\
& 3x_1 & + & x_2 & + & x_3 & & & \geq & 6 \\
& & & x_2 & & & + & 2x_4 & \geq & 3 \\
& & & & & x_3 & & & \geq & 2 \\
& & & & & & & x_j & \geq & 0 \quad \text{and integer.}
\end{array}
\tag{21}
$$

The LP-relaxation of (21) has the optimal solution $(x_1, x_2, x_3, x_4) = (\frac{1}{14}, 3, 2\frac{11}{14}, 0)$ with objective value $5\frac{6}{7}$. Without column 1 the optimal LP solution is $(x_2, x_3, x_4) = (3\frac{1}{4}, 2\frac{3}{4}, 0)$ with value 6.

We shall use $F_1(d) = \lceil \frac{1}{10}d_1 + \frac{3}{10}d_2 + \frac{1}{10}d_3 + \frac{7}{10}d_4 \rceil$ to generate the cutting plane $x_3 \geq 3$. The resulting problem has an optimal and integer solution $(x_2, x_3, x_4) = (3, 3, 0)$, corresponding to the efficient solution $x = (0, 3, 3, 0)$.

We perform sensitivity analysis on the right hand side of the first constraint, i.e. ε, to show that other efficient solutions can be obtained without changing the value of ε and resolving the problem.

(20) without x_1 is

$$
\begin{array}{rrrrrrrl}
\min & x_2 & + & x_3 & + & x_4 & & \\
\text{s.t.} & -4x_2 & & & - & 2x_4 & \geq & -\varepsilon \\
& x_2 & + & x_3 & & & \geq & 6 \\
& x_2 & & & + & 2x_4 & \geq & 3 \\
& & & x_3 & & & \geq & 2 \\
& & & & & x_j & \geq & 0 \quad \text{and integer.}
\end{array}
\tag{22}
$$

F generates the cut $x_3 \geq \lceil -\frac{1}{10}\varepsilon + \frac{35}{10} \rceil$. Observing that the LP relaxation of (20) and (22) is infeasible for $\varepsilon < 3$ we get $x_3 \geq 4$ for $\varepsilon \in [3, 5)$, $x_3 \geq 3$ for $\varepsilon \in [5, 15)$ and a redundant constraint for $\varepsilon \geq 15$.

We add $x_3 \geq 4$ and analyze the (23) for $\varepsilon \in [3, 5)$.

$$
\begin{array}{rrrrrrrl}
\min & x_2 & + & x_3 & + & x_4 & & \\
\text{s.t.} & -4x_2 & & & - & 2x_4 & \geq & -\varepsilon \\
& x_2 & + & x_3 & & & \geq & 6 \\
& x_2 & & & + & 2x_4 & \geq & 3 \\
& & & x_3 & & & \geq & 2 \\
& & & x_3 & & & \geq & 4 \\
& & & & & x_j & \geq & 0 \quad \text{and integer.}
\end{array}
\tag{23}
$$

18

Within that range the dual optimal solution is $(y_1, \ldots, y_4, u_{F_1}) = (\frac{1}{6}, 1, \frac{2}{3},$
$0, 0)$. *Checking the dual constraints we see that* x_1 *will not be generated and using* $F_2(d) = \lceil \frac{1}{3} d_1 + \frac{3}{10} d_3 \rceil$ *and* $F_3(d) = \lceil \frac{1}{6} d_1 + \frac{2}{3} d_3 \rceil$ *we obtain the cuts* $x_2 \le 0$ *and* $x_4 \ge 2$. *With these cuts added, the problem becomes infeasible for* $\varepsilon < 4$ *and for* $\varepsilon \in [4, 5)$ *we have* $(y_1, \ldots, y_4, u_{F_1}, u_{F-2}, u_{F_3}) = (0, 1, 0, 0, 0, 0, 1)$ *so that* $G(d) = d_2 + \lceil \frac{1}{6} d_1 + \frac{2}{3} d_2 \rceil$. *This way, with* $a^1 = (-14, 3, 0, 0)$ *we have* $G(a^1) = 3 > 1$ *and* x_1 *is generated. The coefficients of* x_1 *in the three cuts are* $F_1(a^1) = 0, F_2(a^1) = -4, F_3(a^1) = -2$ *and the resulting LP has the optimal and efficient solution* $x = (0, 0, 6, 2)$, *which is obtained for all* $\varepsilon \in [4, 5)$.

A similar analysis can be done for $\varepsilon \in [5, 15)$ *and* $\varepsilon \ge 15$. *It turns out that for all* $\varepsilon \le 12$ *the dual solution* $(y, u_{F_1}) = (\frac{1}{6}, 1, \frac{2}{3}, 0, 0)$ *remains optimal, so that* x_1 *will never be generated. By adding further cuts, the efficient solutions* $(0, 0, 6, 2)$ *or* $(0, 1, 5, 1)$ *are found for* $\varepsilon \in [5, 6)$ *and* $\varepsilon \in [6, 12)$, *respectively. For* $\varepsilon \ge 12$ *the dual optimal solution changes to* $(y_1, \ldots y_4, u_{F_1}) = (0, 1, 0, 0, 0)$ *and the dual constraints indicate that* x_1 *has to be generated. Then efficient solutions* $(0, 3, 3, 0)$ *and* $(1, 1, 2, 1)$ *are detected for* $\varepsilon \in [12, 20)$ *and* $\varepsilon \in [20, \infty)$, *respectively.*

Below we present an example of the use of our method in a bicriteria set covering problem, using the ε-constraint scalarization and the branch and bound technique

Example 5

$$
\begin{array}{rlrrrrrrrl}
\min & z_1 = 4x_1 & + & 4x_2 & + & 4x_3 & + & 3x_4 & & \\
\text{s.t.} & -x_1 & - & 3x_2 & - & 2x_3 & - & x_4 & \ge & -\varepsilon \\
& x_1 & & & + & x_3 & + & x_4 & \ge & 1 \\
& x_1 & + & x_2 & & & & & \ge & 1 \\
& & & x_2 & + & x_3 & & & \ge & 1 \\
& x_1, & & x_2, & & x_3, & & x_4 & \in & \{0, 1\}.
\end{array}
$$

Here, the first constraint is a constraint derived from a second objective function to minimize $z_2 = x_1 + 3x_2 + 2x_3 + x_4$. *There are two efficient solutions,* $x = (0, 1, 0, 1), z = (z_1, z_2) = (7, 4)$ *and* $x = (1, 0, 1, 0), z = (z_1, z_2) = (8, 3)$.

The LP is infeasible for $\varepsilon < 3$ *and otherwise has optimal solution* $x = (0.5, 0.5, 0.5, 0)$ *with value 6 and optimal dual solution* $y = (0, 2, 2, 2)$ *for all* $\varepsilon \ge 3$. *Thus the column of variable* x_4 *will not be generated for any value of* ε.

Let us consider branching on variable x_2. In the branch $x_2 \geq 1$ we obtain that the LP is infeasible for $\varepsilon < 4$ even if x_4 is generated. For all $\varepsilon \geq 4$ the optimal integer solution $(x_1, x_2, x_3) = (1, 1, 0)$ with duals $(y, w_2) = (0, 4, 0, 0, 4)$ is obtained. Thus, x_4 is generated and introducing it into the problem results in the optimal and integer solution $x = (0, 1, 0, 1)$.

In the branch $x_2 \leq 0$ the LP is infeasible if $\varepsilon < 3$ and again remains infeasible even with x_4. For $\varepsilon \geq 3$ the optimal solution $(x_1, x_2, x_3) = (1, 0, 1)$ with duals $(y, v_2) = (0, 0, 4, 4, 4)$, so x_4 is generated. The updated LP gives optimal solution $x = (1, 0, 1, 0)$.

In practical applications, where very large integer programs have to be solved, the ε-constraint method has numerical disadvantages, because the ε-constraints tend to destroy problem structure. Ehrgott and Ryan (2003) propose a new scalarization technique that has proved to be computationally superior to the ε-constraint method. The elastic constraint scalarization of a multicriteria optimization problem is as follows.

$$
\begin{aligned}
\min \ & \sum_{j=1}^{n} c_i^j x_j + \sum_{j \neq i} p_j s_j \\
\text{s.t.} \ -& \sum_{j=1}^{n} \hat{c}_j x_j + l - s \ = \ -\varepsilon \\
& \sum_{j=1}^{n} a^j x_j \ \geq \ b \\
& \quad\quad\quad x_j \ \geq \ 0, \ \text{integer} \\
& \quad\quad\quad l, s \ \geq \ 0
\end{aligned}
$$

where $l, s \in \mathbb{R}^{p-1}$ are variables of slack and surplus variables, respectively, and $p \in \mathbb{R}^{p-1}$ is a vector of penalties for violating the original ε-constraints in (19). Note that this modification does not change the column generation subproblem

$$
\max_{t \in T} \max_{a,c} \ a\bar{y}^t - \hat{c}\bar{\omega}^t + \bar{w}_j^t - \bar{v}_j^t - c_i^{j'}
$$

in the branch and bound approach, or the column generation approach. Therefore the approaches described above can be applied to the elastic constraint scalarization, too.

6 Conclusion

A main obstacle in integer programming is the high number of variables that are usually required and the difficulty is getting the models solved within a reasonable amount of time. It is therefore of outmost importance to limit the number of variables. Column generation has been used over some time to solve problems with a single objective. For problems with multiple criteria extensive research has been done with continuous, in particular convex models. Here the weighted sums approach plays a major role and different efficient solutions have been obtained by variations in the weights. If integrality is introduced for the variables the weighted sum approach may fail to find interesting efficient alternatives. We have here used the ϵ-constraint approach in which the parameters are changed in order to find alternative efficient solutions. Each time a model is solved we gain dual information which can facilitate the investigation of efficient alternatives and limit the number of integer variables required.

A fundamental concept in multicritiria optimization is efficiency, and the challenging part is to obtain one or multiple efficient solutions. The same concept also applies to data envelopment analysis, in which however the directions for obtaining efficient solutions are given. See for example Cooper *et al.* (2000). In a companion paper (Ehrgott and Tind, 2007) we give an analysis of the so-called free replicability model with integrality conditions proposed by Tulkens Tulkens (1993).

References

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998). Branch and price: Column generation for solving huge integer programs. *Operations Research*, **46**, 316 – 329.

Chankong, V. and Haimes, Y. (1983). *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York.

Cooper, W., Seiford, L., and Tone, K. (2000). *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Kluwer Academic Publishers, Dordrecht.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer, Berlin, 2nd edition.

Ehrgott, M. and Ryan, D. (2003). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, **11**, 139 – 150.

Ehrgott, M. and Tind, J. (2007). Column generation with free replicability in dea. Technical report, Department of Engineering Science, The University of Auckland.

Nemhauser, G. and Wolsey, L. (1988). *Integer and combinatorial optimization*. Wiley, Chichester.

Ralphs, T. and Galati, M. (2006). Decomposition and dynamic cut generation in integer linear programming. *Mathematical Programming*, **106**, 261 – 285.

Schrijver, A. (1986). *Theory of linear and integer programming*. Wiley, Chichester.

Tulkens, H. (1993). On fdh efficiency analysis: Some methodological issues and applications to retail banking, courts, and urban transit. *Journal of Productivity Analysis*, **4**, 183 – 210.

Vanderbeck, F. (2005). *Implementing mixed integer column generation*, pages 331 – 358. Kluwer Academic Publishers, Dordrecht.

Vanderbeck, F. and Wolsey, L. (1996). An exact algorithm for ip column generation. *Operations Research Letters*, **19**, 151 –159.

Villeneuve, D., Desrosiers, J., Lübbecke, M., and Soumis, F. (2005). On compact formulations for integer programs solved by column generation. *Annals of Operations Research*, **139**, 375 – 388.