Column Generation with Free Replicability in DEA

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 27, 2007

Abstract

The evaluation of efficiency scores in data envelopment analysis is based on the construction of artificial decision making units subject to some assumptions, usually requiring convexity of the production possibility set. This demands divisibility in input and output, which is not always possible. The so-called free replicability model, proposed by Henry Tulkens, permits input and output to enter in only discrete amounts. The model is of a mixed integer programming type, for which the number of variables, here corresponding to the decision making units, may be critical in order to reach an optimal solution.

We suggest to use column generation techniques to include only those decision making units that may contribute to the creation of an optimal solution.

Keywords: Data envelopment analysis, free replicable hull, branch and bound, cutting plane, column generation.

The fundamental linear programming models in Data Envelopment Analysis (DEA) have had an enormous impact on theory and application in productivity analysis. They are based on the creation of comparable decision making units (DMUs) subject to some assumptions on disposability, scaling, and aggregation.

The aggregation process usually implies assumptions on convexity, the principal ones being the constant return to scale and the variable return to scale models. The assumptions of convexity have in certain cases been questioned, notably by Tulkens (1993), who introduced the so-called free disposable hull and free replicability hull models. Later discussions and generalizations have been considered by Bogetoft (1996), Bogetoft *et al.* (2000), Kuosmanen (2003) and Post (2001).

This paper shall focus on the free disposable hull model by Tulkens. It is formulated as a mixed integer programming model of a rather general form having one continuous variable. The remaining variables are integer corresponding to each decision making unit in the model. Mixed integer programming is notoriously hard, in particular in the presence of many integer variables. Many books have been written on the subject, see for example Wolsey (1998). An important tool in solving large optimization problems, including problems in integer programming, is column generation. In integer programming this is of particular relevance when many variables are present and perhaps only known implicitly. We shall here consider column generation, which in the current context is relevant in the presence of many decision making units. But it also applies to the case, where a new decision making unit can be created by a separate model, in column generation terminology by a subroutine. The principle idea is that a master problem with a selected number of decision making units quote prices on input and output vectors to be used in the generation of additional decision making units. In this way column generation skips irrelevant DMUs and includes only those that may improve the efficiency score. This discussion is connected to the analysis in Wilson (1995) about the detection of influential observations in data envelopment analysis.

DEA and multicriteria optimization both deal with the concept of an efficient border and share many properties that can be exploited using equivalent techniques. See for example Joro *et al.* (1998). The present paper has a companion paper, Ehrgott and Tind (2007), dealing with multicriteria optimization models.

Section 1 presents the fundamental model to be studied. Section 2 describes how a fundamental branch and bound scheme can be carried out for the solution of the model. This is accompanied by an example. Section 3 deals with the introduction of possibly advantageous DMUs to be included in the branch and bound procedure followed by some discussion including an economic interpretation.

A similar analysis is carried out with the other fundamental solution method in integer programming, the cutting plane technique, in Section 4. The paper concludes with some perspectives about the common features of the two procedures in the present context.

1 The Free Replicability Model

The free replicability model considers integer combinations of the decision making units, DMUs. Let us introduce some notation in order to state the model. Let x^j and y^j denote the input and the output vector of the jth DMU. Consider a particular DMU with input vector x^0 and output vector y^0. Let $j = 0, \ldots, n$, i.e. we have $n+1$ DMUs. Let θ, λ_j be variables, $j = 0, \ldots, n$. The free replicability model can then be formulated in the following way.

$$
(1) \qquad \begin{array}{lrcll}
\min & \theta & & & \\
\text{s.t.} & \theta x^0 - \sum_{j=0}^{n} x^j \lambda_j & \geq & 0 & \\
& \sum_{j=0}^{n} y^j \lambda_j & \geq & y^0 & \\
& \lambda_j & \geq & 0 & (j = 0, \ldots, n) \\
& \theta & & \text{free.} &
\end{array}
$$

This is a mixed integer programming problem with integer variables and (here) one continuous variable. For simplicity we have omitted the handling of slacks. However, they may be introduced without difficulty. For the treatment of slacks see Cooper *et al.* (2000).

In general it is difficult to solve such problems if there are many integer variables, i.e. if there are many DMUs. We shall therefore use a column generation technique in order to exclude DMUs that cannot contribute to the formation of an optimal solution of (1).

Additionally, column generation techniques allow for the generation of DMUs that are not given directly, but for which there are certain restrictions on a feasible input-output combination (x^j, y^j). In general this could be given as a production possibility set P such that $(x^j, y^j) \in P$. Restrictions could also, perhaps more naturally, be given as a set of constraints $g(x^j, y^j) \leq 0$ defined by limitations in production factors.

Column generation has widespread use in linear programming, and in many cases also in integer or mixed integer programming where only the linear programming relaxation is considered and solved via column generation. The trouble is, however, that some columns, here corresponding to the DMUs, may not be considered in a linear programming relaxation, even though they are part of an optimal solution.

We shall here outline a column generation method for (1). For (mixed) integer programming problems there exist generally two basic solution principles, branch and bound techniques and cutting plane techniques, and column generation techniques are also developed in (mixed) integer programming, see Barnhart *et al.* (1998) and Vanderbeck and Wolsey (1996), in particular for the branch and bound method. We shall apply both methods on the free replicability model in a column generation framework.

2 Branch and Bound

As mentioned, the idea is to start the process with a smaller number k of known DMUs, where $k < n + 1$. In branch and bound we consider in a tree structure multiple linear programming relaxations of the original problem (1). If fractional values of integer variables are present in an optimal solution of a relaxed problem, this problem is typically split into multiple problems by the exclusion of the open non-integer interval around the fractional values of the integer variables. Typically this is only done with respect to a single variable, and in this way a problem is replaced by two new linear programming problems in a recursive way. We shall here follow this approach. Each problem will have the same structure but with different bounds put on the integer variables. Let l_j and u_j denote the lower and upper bounds, respectively, of the variables λ_j for $j = 0, \ldots, k$. With this notation each problem has the following general structure.

$$
\begin{array}{llll}
\min & \theta & & \\
\text{s.t.} & \theta x^0 - \sum_{j=0}^{k} x^j \lambda_j \geq 0 & & \\
 & \sum_{j=0}^{k} y^j \lambda_j \geq y^0 & & \\
 & \lambda_j \leq u_j & & (j = 0, \ldots, k) \\
 & \lambda_j \geq l_j & & (j = 0, \ldots, k) \\
 & \theta \quad \text{free.} & &
\end{array}
\tag{2}
$$

The first problem to be considered is the linear programming relaxation of (1). Here in the beginning $l_j = 0$ and $u_j = M$ for $j = 0, \ldots, k$, where M is a nonbinding number. We shall here illustrate the process on an example. For more information and details about branch and bound see for example Wolsey (1998).

Example 1 *Let us consider a problem with 5 DMUs: A, B, C, D and E having 1-dimensional input and one 1-dimensional output as given by the following table.*

DMU	*A*	*B*	*C*	*D*	*E*
input x^j	6	2	5	3	8
output y^j	12	3	4	5	7

Let C be the designated DMU to be evaluated. Let us also assume that at the beginning we have no knowledge about DMUs D and E. With this terminolgy our problem (1) takes the following form

$$\begin{array}{llrrrrrrrl}
\min & \theta & & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & \geq & 0 \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & \geq & 4 \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C & \geq & 0 \textit{ and integer} \\
 & \theta & & & & & & & & \textit{free.}
\end{array}$$

Let us first look at the linear programming relaxation with the unrestrictive bound $u_j = 10$.

Problem I:

$$\begin{array}{llrrrrrrrl}
\min & \theta & & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & \geq & 0 \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & \geq & 4 \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C & \geq & 0 \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C & \leq & 10 \\
 & \theta & & & & & & & & \textit{free.}
\end{array}$$

The optimal solution of this program gives us $\theta = 0.4$ for $\lambda_A = 0.33$ and $\lambda_B = \lambda_C = 0$. Since λ_A is fractional we remove the open interval (0,1) from consideration for λ_A. Hence problem I is replaced by the next two problems II and III.

Problem II:

$$\begin{array}{llrrrrrrrl}
\min & \theta & & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & \geq & 0 \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & \geq & 4 \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C & \leq & 10 \\
 & & & \lambda_A & & & & & \geq & 1 \\
 & & & & & \lambda_B, & & \lambda_C, & \geq & 0 \\
 & \theta & & & & & & & & \textit{free.}
\end{array}$$

The optimal solution of problem II is $\lambda_A = 1$, $\lambda_B = \lambda_C = 0$ with $\theta = 1.2$. This solution is integer.

Problem III:

$$\begin{array}{lcccccccl}
\min & \theta & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & \geq 0 \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & \geq 4 \\
 & & & \lambda_A, & & \lambda_B & & \lambda_C & \geq 0 \\
 & & & \lambda_A & & & & & \leq 0 \\
 & & & & & \lambda_B, & & \lambda_C & \geq 0 \\
 & \theta & & & & & & & \textit{free.}
\end{array}$$

The optimal solution of problem III is $\lambda_A = \lambda_C = 0, \lambda_B = 1.33$ *with* $\theta = 0.533$. *This solution is still fractional. Hence problem III is replaced by the next two problems IV and V. We provide only the bounding constraints.*

Problem IV:

$$0 \geq \lambda_A \geq 0,\ 1 \geq \lambda_B \geq 0,\ 10 \geq \lambda_C \geq 0.$$

The optimal solution of problem IV is $\lambda_A = 0$ $\lambda_B = 1$ *and* $\lambda_C = 0.25$ *with* $\theta = 0.65$.

Problem V:

$$0 \geq \lambda_A \geq 0,\ 10 \geq \lambda_B \geq 2,\ 10 \geq \lambda_C \geq 0.$$

The optimal solution of problem V is $\lambda_A = \lambda_C = 0, \lambda_B = 2$ *with* $\theta = 0.8$. *This solution is integer and better than the solution provided by problem II. Hence problem II is left from further consideration. Finally, problem IV is separated into the next two problems VI and VII.*

Problem VI:

$$0 \geq \lambda_A \geq 0,\ 1 \geq \lambda_B \geq 0,\ 0 \geq \lambda_C \geq 0.$$

Problem VI is infeasible.

Problem VII:

$$0 \geq \lambda_A \geq 0,\ 1 \geq \lambda_B \geq 0,\ 10 \geq \lambda_C \geq 1.$$

The optimal solution of problem VII is $\lambda_A = \lambda_B = 0$ and $\lambda_C = 1$ with $\theta = 1.0$, which is integer, but not better than the solution provided by problem V.

There are no remaining problems to be considered. Hence the solution $\lambda_A = \lambda_C = 0, \lambda_B = 2$ with $\theta = 0.8$ obtained by problem V is optimal in this example. For the development of the tree structure in the branch and bound process see Figure 1.

< FIGURE 1 HERE.>

So DMU C is not efficient, as it obtains a productivity index which is lower than 1. A new DMU twice as large as DMU B can obtain at least the required output with a smaller amount of input. Observe that with the required integrality the efficiency has improved in comparison with the value $\theta = 0.4$ of the linear programming relaxation, where DMU C became compared with a fractional part of DMU A. △

3 Column Generation in Branch and Bound

Let us again consider the general problem (2) and its dual linear programming problem

$$(3) \quad \begin{array}{lrcll} \max & vy^0 + \sum_{j=0}^{k}(w_j l_j - z_j u_j) & & & \\ \text{s.t.} & ux^0 & \leq & 1 & \\ & -ux^j + vy^j + w_j - z_j & = & 0 & (j = 0, \dots, k) \\ & u & \geq & 0 & \\ & v & \geq & 0 & \\ & w_j, z_j & \geq & 0 & (j = 0, \dots, k), \end{array}$$

where u, v, w_j, z_j are dual variables of appropriate dimensions.

We shall here apply the column generation technique or decomposition technique from linear programming. This tells us that problem (2) is solved to optimality even with all DMUs included, if the inequality

$$(4) \qquad -ux^j + vy^j + w_j - z_j = 0$$

is satisfied for all $j = 0, \ldots, n$, where (u, v, w_j, z_j) are optimal in (3). Since possible new columns have no restrictive upper bound at the outset the z_j variable is not required for this check. Also at the outset the lower bound is equal to 0, which has the effect that w_j may be removed and the equation (4) may be turned into the following inequality

$$(5) \qquad -ux^j + vy^j \leq 0.$$

Example 2 *Let us return to Example 1. Consider again Problem I, which is just the linear programming relaxation of the original problem. For this problem the optimal dual variables are* $u = 0.2, v = 0.1$. *We shall now check whether the dual inequality (5) is satisfied for DMU D with these dual variables. Calculation shows that*

$$-0.2 \times 3 + 0.1 \times 5 \leq 0.$$

Hence DMU D should not be considered if we are only going to study the linear programming relaxation of our problem. △

Going back to the general problem the branch and bound procedure terminates with a set of problems that do not need to be investigated further. Either a problem is infeasible or the objective value, which is a lower bound for all possible further separations, is not lower than the value of the best integer solution obtained.

However, it could be that a new column may change this picture. It is here column generation comes into place. This implies that the inequalily (5) should be checked for all problems that are not yet separated, also called the terminal problems.

If the inequality does not hold for a new column for a particular problem, then that column or DMU should be introduced into the problem, which should be solved again. This may decrease the value of the problem and hence give rise the further calculations by the branch and bound procedure. Similarly, if a terminal problem is infeasible a new column should be introduced into that problem. Since the variable of a new column at the outset has no upper bound and since all input and output data are usually assumed strictly positive, then feasibility can be obtained for a sufficiently high λ-value of the new column. When all terminal problems are feasible and the inequality (5) is satisfied for all new columns then the entire procedure terminates with an optimal solution of the original problem.

Example 3 *We shall continue with the previous example. The terminals after termination of the branch and bound procedure correspond to problems II, V, VI, and VII. Let us check the inequalities for DMU D in problem II. The optimal dual variables are here* $(u, v) = (0.2, 0)$ *and the inequality (5)*

$$-0.2 \times 3 + 0 \times 5 \leq 0.$$

is thus satisfied. Similarly for problems V and VII.

However, for the infeasible problem VI we insert DMU D and we obtain Problem VIII.

Problem VIII:

$$\begin{array}{llccccccccccl}
\min & \theta & & & & & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & - & 3\lambda_D & \geq & 0 & \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & + & 5\lambda_D & \geq & 4 & \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C, & & \lambda_D & \geq & 0 & \\
 & & & \lambda_A, & & & & \lambda_C, & & & \leq & 0 & \\
 & & & & & \lambda_B & & & & & \leq & 1 & \\
 & & & & & & & & & \lambda_D & \leq & 10 & \\
 & \theta & & & & & & & & & & & \textit{free.}
\end{array}$$

As the optimal solution has the fractional element $\lambda_D = 0.8$ *the problem is separated into two problems IX and X.*

Problem IX *with the bounding constraints*

$$0 \geq \lambda_A \geq 0, \; 1 \geq \lambda_B \geq 0, \; 0 \geq \lambda_C \geq 0, \; 10 \geq \lambda_D \geq 1.$$

This has the optimal integer solution $\lambda_A = \lambda_B = \lambda_C = 0, \lambda_D = 1$ *and* $\theta = 0.6$*, improving the previous best solution forund in problem V. The opposite*

Problem X *has the bounding constraints*

$$0 \geq \lambda_A \geq 0,\ 1 \geq \lambda_B \geq 0,\ 0 \geq \lambda_C \geq 0,\ 0 \geq \lambda_D \geq 0.$$

This problem is infeasible. Hence we shall introduce DMU E into this problem. However this gives the value $\theta = 0.62$ *which exceeds the values of all terminals.*

Finally, DMU E satisfies the inequality (5) in all terminals, and the procedure terminates with the solution found in problem IX as optimal for the entire problem. See also Figure 2. This shows that DMU D performs even better than a double sized DMU B.

< Figure 2 Here. >

△

Discussion. The above scheme has been built upon the traditional branch and bound approach. If implemented directly it is necessary to have explicit knowledge about included DMUs. It may happen that the dual feasiblity condition (5) suggests introduction of a DMU that is already in. This situation may occur if the dual variable z_j is positive for an already included DMU. Other branching rules have been suggested to overcome this difficulty via the introduction of different branching rules. See for example Vanderbeck and Wolsey (1996) and Barnhart *et al.* (1998).

It should also be noted that columns (DMUs) may be created in a separate production program of the following general form. We omit the index j.

$$\begin{aligned} &\max -ux + vy \\ &\quad \text{s.t. } (x, y) \in P. \end{aligned}$$

If the value of this program exceeds 0 then a the generated DMU is inserted. Otherwise the procedure stops, provided that also possible infeasible terminals have been explored.

The entire procedure has a straightforward economic interpretation. Let T be the index set of all terminals at the termination of the procedure and let (u_t, v_t) denote the optimal dual variables associated with terminal $t \in T$. Termination of the procedure requires that $-u_t x^j + v_t y^j \leq 0$ for all DMUs, i.e. for $j = 0, \dots, n$ and all $t \in T$.

Let $G(-x^j, y^j) = \max_{t \in T} -u_t x^j + v_t y^j$. The termination criterion can then be stated in short form as

$$G(-x^j, y^j) \leq 0 \qquad (j = 0, \dots, n).$$

The function G can be considered as a revenue function taking in costs for consumption of inputs and income from outputs. The termination criterion then says, that no DMU can have positive revenue. Otherwise and according to the procedure above, if a DMU violates the termination criterion there may by a gain in the efficiency score by introducing that DMU.

4 Column Generation and Cutting Planes

In this section we shall first consider the case with 1-dimensional input, i.e. $x^j \in \mathbb{R}_+$. In this case the program (1), apart from the scaling of the objective, is equivalent to the following pure integer programming problem.

$$\text{(6)} \qquad \begin{aligned} \min \sum_{j=0}^{n} x^j \lambda_j & \\ \text{s.t.} \sum_{j=0}^{n} y^j \lambda_j &\geq y^0 \\ \lambda_j &\geq 0 \text{ and integer } (j = 0, \dots, n). \end{aligned}$$

From the theory of integer programming the explicitly stated integrality condition of the above problem may be eliminated by the introduction of extra constraints, the so-called cutting planes, which cut away nonintegral corner points of the constraints in the linear programming relaxation. We shall here introduce the fundamental Chvatal-Gomory (C-G) cuts. These cuts are derived from the original constraints through a recursive use of the following operations on the rows:

- Addition,
- multiplication by a non-negative scalar,
- application of the round-up operation $\lceil \cdot \rceil$. For example $\lceil 5.4 \rceil = 6$.

More on the use of cutting plane techniques in integer programming can be found in Wolsey (1998).

Example 4 *We shall continue with the previous example and as before we shall consider the linear programming relaxation of the reduced problem without DMU D:*

$$
\begin{array}{llcrcrcrcl}
\min & \theta & & & & & & & & \\
s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & \geq & 0 \\
 & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & \geq & 4 \\
 & & & \lambda_A, & & \lambda_B, & & \lambda_C & \geq & 0 \\
 & \theta & & & & & & & & \textit{free.}
\end{array} \tag{7}
$$

First multiply the second constraint in (7) by $\frac{1}{3}$ and obtain

$$4\lambda_A + \lambda_B + \frac{4}{3}\lambda_C \geq \frac{4}{3}.$$

Afterwards apply the round-up operation on all cooefficients. In this way we obtain the first cut.

First cut:

$$4\lambda_A + \lambda_B + 2\lambda_C \geq 2. \tag{8}$$

Similarly, multiply again the second constraint of the problem but this time by $\frac{1}{12}$. After rounding up of all coefficients we get the second cut.

Second cut:

$$\lambda_A + \lambda_B + \lambda_C \geq 1. \tag{9}$$

Add (8) and (9) and multiply the result by $\frac{1}{2}$ and obtain

$$\frac{5}{2}\lambda_A + \lambda_B + \frac{3}{2}\lambda_C \geq \frac{3}{2}.$$

The coefficients are rounded up and we get the following third cut.

Third cut:

$$3\lambda_A + \lambda_B + 2\lambda_C \geq 2. \tag{10}$$

Add next the inequalities (9) and (10). Multiply the resulting inequality by $\frac{1}{2}$ and we get

$$2\lambda_A + \lambda_B + \frac{3}{2}\lambda_C \geq \frac{3}{2}$$

and upon round-up we obtain the fourth cut.

Fourth cut:

$$2\lambda_A + \lambda_B + 2\lambda_C \geq 2. \tag{11}$$

We shall keep our attention to the cut (11). The cuts (8) and (10) are dominated by cut (11). For simplicity of exposition we also leave out (9) from further consideration, as we know in this example that it is going to be nonbinding. With the notation already introduced let $y^j \in \mathbb{R}$ denote the jth coefficient in the second row of (7) and define the function

$$F(y^j) = \left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{3}y^j \right\rceil + \left\lceil \frac{1}{12}y^j \right\rceil\right)\right\rceil + \left\lceil \frac{1}{12}y^j \right\rceil\right)\right\rceil. \tag{12}$$

The formula (12) just gives a direct calculation of the jth coefficient in the cut (11) based on the above detailed development.

If we solve (7) with the addition of the cut (11) we get the optimal and integer solution $\lambda_A = \lambda_C = 0, \lambda_B = 2$ with $\theta = 0.8$, as wanted. So the cut (11) is enough to cut off undesired fractional solutions. $\triangle$

In general we get multiple cuts, in which the coefficients can be calculated based on a formula subject to the generating rules of the C-G cuts. If we have m outputs and l cuts we may naturally consider the vector function $F(y^j) : \mathbb{R}^m \to \mathbb{R}^l$. Let $w \in \mathbb{R}^l$ denote the optimal dual variables of the cuts. Together with previous notation the dual feasibility may be stated as

$$-ux^j + vy^j + wF(y^j) \le 0. \tag{13}$$

By optimality this is valid for all the DMUs introduced, i.e. for $j = 0, \ldots, k$. The question is whether dual feasibility holds for all DMUs.

Example 5 *In the example we get the optimal dual variables $(u, v, w) = (0.2, 0, 0.4)$, and as an illustration we check dual feasibility (13) for DMU A by calculating*

$$-0.2 \times 6 + 0 \times 12 + 0.4 \times 2 = -0.4 \le 0.$$

This also applies to the remaining included DMUs, B and C.

The question is whether this condition is also true for DMU D. Let us first calculate the corresponding coefficient in the cut by means of (12).

$$\begin{aligned} F(5) &= \left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{3} \times 5 \right\rceil + \left\lceil \frac{1}{12} \times 5 \right\rceil\right)\right\rceil + \left\lceil \frac{1}{12} \times 5 \right\rceil\right)\right\rceil \\ &= \left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{2}(2+1) \right\rceil + 1\right)\right\rceil \\ &= \left\lceil \frac{1}{2}(2+1) \right\rceil \\ &= 2. \end{aligned}$$

Now let us check condition (13) for DMU D. Calculation shows

$$-0.2 \times 3 + 0 \times 5 + 0.4 \times 2 = 0.2.$$

Hence condition (13) is violated and we shall introduce DMU D in our LP relaxation including the added and now extended cut. Hence we shall solve

$$\begin{array}{llrrrrrrrrl} \min & \theta & & & & & & & & & \\ s.t. & 5\theta & - & 6\lambda_A & - & 2\lambda_B & - & 5\lambda_C & - & 3\lambda_D & \ge 0 \\ & & & 12\lambda_A & + & 3\lambda_B & + & 4\lambda_C & + & 5\lambda_D & \ge 4 \\ & & & 2\lambda_A & + & \lambda_B & + & 2\lambda_C & + & 2\lambda_D & \ge 2 \\ & & & \lambda_A, & & \lambda_B, & & \lambda_C, & & \lambda_D & \ge 0 \\ & \theta & & & & & & & & & \textit{free.} \end{array}$$

The optimal solution of this problem is $\lambda_A = \lambda_B = \lambda_C = 0, \lambda_D = 1$, *which is integer. The corresponding dual solution is* $(u, v, w) = (0.2, 0, 0.3)$.

Let us finally consider DMU E and we shall therefore calculate the corresponding coefficient in the cut by means of (12).

$$\begin{aligned} F(7) &= \left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{3}\times 7\right\rceil + \left\lceil \frac{1}{12}\times 7\right\rceil\right)\right\rceil + \left\lceil \frac{1}{12}\times 7\right\rceil\right)\right\rceil \\ &= \left\lceil \frac{1}{2}\left(\left\lceil \frac{1}{2}(3+1)\right\rceil + 1\right)\right\rceil \\ &= \left\lceil \frac{1}{2}(2+1)\right\rceil \\ &= 2. \end{aligned}$$

Now let us check condition (13) for DMU E. Calculation shows

$$-0.2 \times 8 + 0 \times 7 + 0.3 \times 2 = -1.0.$$

Hence (13) is satisfied and the procedure terminates with the last solution with value $\theta = 0.6$. △

Discussion. The formation of cuts can be done in many ways. Many integer programming textbooks include the generation of the so-called fractional Gomory cuts, which over a long period of time were considered of limited practical use. However, they have been revived in recent years and have been adopted in modern optimization software.

In the present context it should be noted that we will ensure finite convergence of the above column generation procedure by using Gomory cuts with integral data.

It should also here be noted that the generation of DMUs may be created by a production program, which here will have the following general form. We omit the index j.

$$\begin{aligned} \max -ux + vy + wF(y) & \\ \text{s.t. } (x, y) &\in P. \end{aligned}$$

So far, we have limited ourselves to the discussion of the case with only 1-dimensional input, since the problem in this instance could be transformed into an ordinary pure integer programming problem (6). With multi-dimensional input the original problem (1) remains a genuine mixed integer programming problem. Cut generation in mixed integer programming is a much harder discipline, although some very nice theoretical results exist together with promising computational practice. See for example Cornuéjols (2007). Gomory introduced at an early stage cuts for mixed integer programming which have been treated in many textbooks. See for example Nemhauser and Wolsey (1988). It is possible in the mixed integer programming case to build up functions like the above ones in order to generate coefficients for a cut. They can be cast into the general form $G(-x^j, y^j)$, and dual feasibility can be checked by the sign of this function. For some details see Agrell and Tind (2001).

As in the branch and bound case we can also give an economic interpretation. Let $G(-x, y) = -ux + vy + wF(y)$. Then the dual feasibility condition can be given the short form $G(-x^j, y^j) \leq 0$ for $j = 0, \ldots, n$. If G is considered as a revenue function this says that there is no room for an additional gain in the efficiency score from any DMU at the end of the whole procedure.

5 Conclusion

We have shown how some DMUs may be selected for and other DMUs may be left out of consideration in an optimal solution. This has here been demonstrated using both of two fundamental solution tools in integer programming. In particular it is noted that the linear programming relaxation of the original problem cannot alone provide the correct dual variables in order to obtain the relevant DMUs.

The two cases, branch and bound, and cutting planes, have been considered separately. However, we have seen that the rules for insertion along with an economic interpretation can be cast in the same framework by means of the general function G. Hence, it should be noted that it is possible to merge the two procedures within a common framework for the study of the free replicability hull model (1).

In the classical linear programming case primal and dual feasibility together with the notion of complementary slackness play a key role in sensitivity analysis studies of the classical models. See for example Cooper *et al.* (2001) and Boljunčić (2006). As a final comment we may observe that the notion of primal and dual feasibility and complementary slackness can be extended into the current mixed integer programming case. This opens up for a sensitivity analysis study of the free replicability model (1) as well.

References

Agrell, P. J. and Tind, J. (2001). A dual approach to nonconvex frontier models. *Journal of Productivity Analysis*, **16**, 129 – 147.

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

Bogetoft, P. (1996). DEA on relaxed convexity assumptions. *Management Science*, **42**, 457 – 465.

Bogetoft, P., Tama, J. M., and Tind, J. (2000). Convex input and output projections of nonconvex production possibility sets. *Management Science*, **46**, 858 – 869.

Boljunčić, V. (2006). Sensitivity analysis of an efficient DMU in DEA model with variable returns to scale (VRS). *Journal of Productivity Analysis*, **25**, 173 – 192.

Cooper, W. W., Seiford, L. M., and Tone, K. (2000). *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software*. Kluwer Academic Publishers, Dordrecht.

Cooper, W. W., Li, S., Seiford, L. M., Tone, K., Thrall, R. M., and Zhu, J. (2001). Sensitivity and stability analysis in DEA: Some recent developments. *Journal of Productivity Analysis*, **15**, 217 – 246.

Cornuéjols, G. (2007). Valid inequalities for mixed integer linear programs. *Mathematical Programming, Series B*. To appear.

Ehrgott, M. and Tind, J. (2007). Column generation in integer programming with applications in multicriteria optimization. Technical report, Department of Engineering Science, The University of Auckland.

Joro, T., Korhonen, P., and Wallenius, J. (1998). Structural comparison of data envelopment analysis and multiple objective linear programming. *Management Science*, **44**, 962 – 970.

Kuosmanen, T. (2003). Duality theory of non-convex technologies. *Journal of Productivity Analysis*, **20**, 273 – 304.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley, New York.

Post, T. (2001). Estimating non-convex production sets using transconcave DEA. *European Journal of Operational Research*, **131**, 132 – 142.

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

Wilson, P. W. (1995). Detecting influential observations in data envelopment analysis. *Journal of Productivity Analysis*, **6**, 27 – 45.

Wolsey, L. A. (1998). *Integer Programming*. Wiley-Interscience, Chichester.

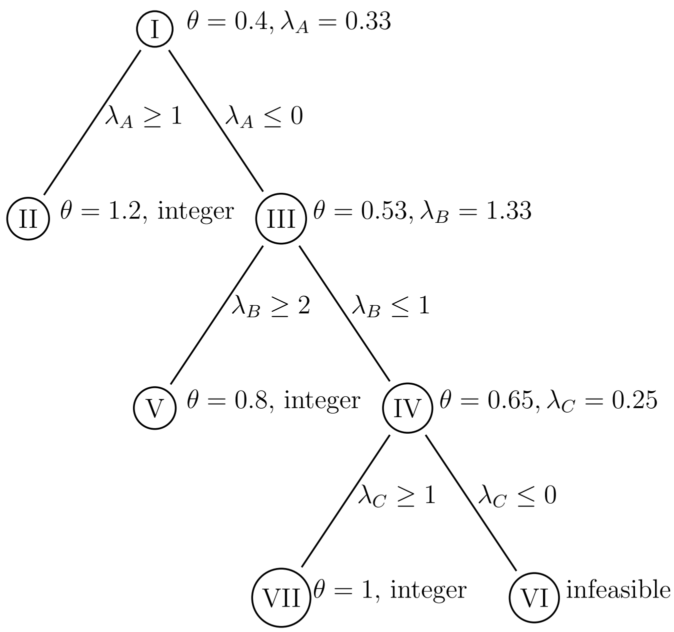

Figure 1: Branch and bound tree.

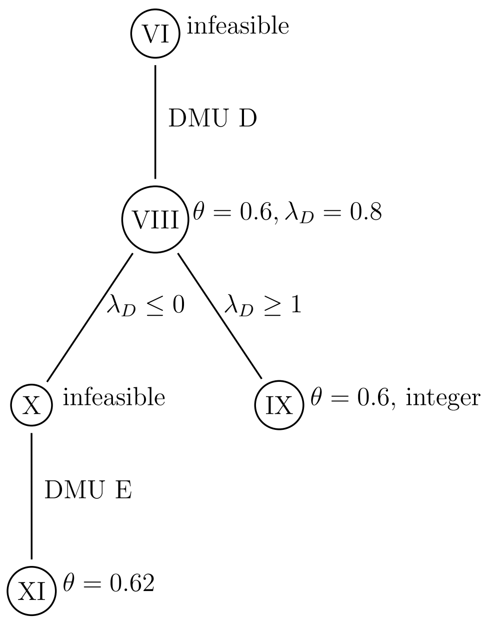

Figure 2: Continuation of branch and bound procedure.