MIXED INTEGER LINEAR FRACTIONAL PROGRAMMING BY A BRANCH AND BOUND TECHNIQUE

ILDIKÓ ZSIGMOND
Computing Center of Eötvös Loránd University
1117 Budapest, Bogdánfy u. 10/b.

(Received May 12, 1984 – revised October 10, 1985)

The branch and bound principle as a practical computational procedure for mixed integer linear programming was first proposed by Land and Doig [10]. Since then the algorithm and its computational efficiency has been considerably improved. All the commercial codes for mixed integer linear programming seem now to be based on this approach (see the survey by Land and Powell [11]).

The algorithm proposed in the present paper is an adaptation of this well-known method to the case of linear fractional objective function. On the basis of the close relationship between the continuous linear and linear fractional programs it is shown that solving the integer constrained versions of these problems is essentially also equivalent. The branch and bound tree can be developed in the same way using any branching strategy applicable for the linear case, and equivalent, but differently computable penalties can be imposed on the generated subproblems.

1. Introduction

Optimization problems involving ratios in the objective function – commonly called fractional programs – have been studied in a considerable number of papers. A comprehensive bibliography of more than 550 publications has been recently published by Schaible [14].

The present paper deals with linear fractional programs (LFP) with additional integrality constraints imposed on all or some of the variables.

In the last 10 – 15 years several papers treating this problem have been published. Some of them deals with pure integer or specially with (0, 1) fractional programs. The main approaches are Boolean methods, iterative parametric methods, cutting planes, implicit enumeration with surrogate constraints, branch and bound methods etc. For further details we refer to Schaible's bibliography [14].

In the general mixed integer case (MILFP), like in mixed integer linear programming (MILP), the branch and bound (B&B) approach seems to be the most promising one. Bitran and Magnanti [7] describe a parametric primal-dual algorithm for LFP, with a proposition to apply this algorithm for solving the subproblems and for developing the fathoming tests in a B&B algorithm for MILFP. B&B approaches have been proposed also by Agrawal [2, 3], Chandra and Chandramohan [8], Agrawal and Chand [4]. Of these only [8] utilizes the dual of LFP to calculate penalties due to Beale and Small [5]. This approach applies the Charnes-Cooper transformation [9] on variables, therefore the branching rules are not straightforward, and the simple lower and upper bounds of the variables are treated as additional rows in the simplex tableau.

On the basis of the dual algorithm proposed in the present paper it is possible not only to calculate the above-mentioned penalties, but also all the penalty improvements proposed for the linear case by Tomlin [15]. Furthermore, although we also utilize implicitly the Charnes-Cooper linear programming (LP) equivalent of LFP, the problem and its variables are kept in their original form, and the simple bounds on the variables are treated implicitly.

The paper is organized as follows: Section 2 provides the notations and definitions. In Section 3 a dual algorithm for LFP is outlined. Section 4 deals with the calculation of penalties. Section 5 contains some conluding remarks, and finally, in Section 6 a numerical example, elaborated in detail, illustrates the algorithm.

The list of references is not meant to be comprehensive; only papers closely related to the present algorithm are included.

2. Notations, definitions

The mixed integer linear franctional programming problem can be formulated as

$$\text{(2.1)} \qquad \text{MILFP:}\ z_x = \max\left\{\frac{f(x)}{g(x)} = \frac{c^T x+\alpha}{d^T x+\beta} \,\middle|\, x \in X^0 \cap S\right\},$$

where

$$\text{(2.2)} \qquad X^0 = \{x \in \mathbf{R}^n \,|\, Ax \leq b,\ L^0 \leq x \leq U^0),$$

and

$$\text{(2.3)} \qquad S = \{x \in \mathbf{R}^n \,|\, x_j \text{ integer for } j \in J\}.$$

Here A is an $(m \times n)$ matrix of real numbers, $c, d, L^0, U^0 \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, $\alpha, \beta \in \mathbf{R}$, $J \subseteq N = \{1, 2, \ldots, n\}$ is the index-set of the variables constrained to take on integer values, the superscript T denotes the transpose, and other superscripts are subproblem indices.

Let us denote the continuous relaxation of MILFP by LFP^0:

$$\text{(2.4)} \qquad \text{LFP}^0\text{:}\ z^0 = \max\left\{\frac{c^T x+\alpha}{d^T x+\beta} \,\middle|\, x \in X^0\right\}.$$

For LFP^0 the following (common) assumptions are made:

A1: $X^0 \neq 0$ and bounded,

A2: $d^T x + \beta > 0$ for all $x \in X^0$.

An additional, but not essential assumption on MILFP:

*A*3: $L_j^0 \leq U_j^0$ are finite integer numbers for $j \in J$.

The B&B method, as applied to our problem, involves temporarily ignoring the constraint $x \in S$ and solving a sequence of continuous subproblems. These all have the same form as LFP^0, except that they have different sets of lower and upper bounds L^k and U^k.

Let the k-th subproblem be defined as

$$\text{(2.5)} \qquad LFP^k\colon\ z^k = \max\left\{\frac{c^T x + \alpha}{d^T x + \beta}\ \middle|\ x \in X^k\right\},$$

where

$$\text{(2.6)} \qquad X^k = \{x \in X^0 \mid L^0 \leq L^k \leq x \leq U^k \leq U^0\},$$

at least one of the inequalities $L^0 \neq L^k$, $U^0 \neq U^k$ is satisfied, and L^k and U^k are integer numbers.

Denote the Charnes-Cooper equivalent of LFP^k by LP^k:

$$\text{(2.7)} \qquad LP^k\colon\ \text{maximize } c^T y + \alpha t$$

subject to

$$\text{(2.8)} \qquad Ay - bt \leq 0$$

$$\text{(2.9)} \qquad -y + L^k t \leq 0$$

$$\text{(2.10)} \qquad y - U^k t \leq 0$$

$$\text{(2.11)} \qquad d^T y + \beta t = 1$$

$$\text{(2.12)} \qquad y \geq 0,\ t \geq 0,$$

where LP^k is obtained from LFP^k by transforming the variables according to $t = 1/(d^T x + \beta)$, $y = tx$.

For notational convenience the coefficient matrix $[A, I]$ of the inequality $Ax \leq b$ is rearranged and partitioned as $[A_B, A_N]$, where A_B is the matrix of a basis. $[c^T, 0^T]$, $[d^T, 0^T]$ and $[x^T, (b - Ax)^T]$ are partitioned conformally, introducing the slack variables as part of the vectors x_B or x_N. Analogous notations for the elements of U, L, y etc. will be used throughout the paper.

3. A dual algorithm for solving the subproblems

The algorithm proposed here is in fact a technique to solve (at least partially) the dual of LFP^k (i. e. that of LP^k) in terms of the original x variables. More precisely, a sequence of simplex tableaux analogous to those of

Martos' primal method is derived, such that the corresponding solutions are dual feasible, and the value of the dual objective function is nonincreasing.

The dual of LFP^k can be formulated as an ordinary LP problem. (About duality models of LFP see e. g. [1, 7, 13].)
Let it be denoted by DLP^k:

$$\text{(3.1)} \qquad DLP^k\text{: minimize } v$$

subject to

$$\text{(3.2)} \qquad u^T A - w_L^T + w_U^T + v d^T \geq c^T$$

$$\text{(3.3)} \qquad -u^T b + w_L^T L^k - w_U^T U^k + v\beta \geq \alpha$$

$$\text{(3.4)} \qquad u \geq 0, \; w_L \geq 0, \; w_U \geq 0,$$

where (since DLP^k is actually the dual of LP^k) the dual variables $u \in \mathbf{R}^m$ belong to the constraints (2.8), $v \in \mathbf{R}$ to (2.11). $w_L \in \mathbf{R}^n$ belongs to (2.9) and $w_U \in \mathbf{R}^n$ to (2.10), i. e. to the constraints of LFP^k: $x \geq L^k$ and $x \leq U^k$, respectively.

It is easy to see that if the current value of the denominator for a (not necessarily primal feasible) basic solution of LFP^k is not zero, then (3.3) in DLP^k is satisfied as an equality. (This follows from the theorem of complementary dual variables of linear programming, since in this case the equivalent basis of LP^k must contain t as a basic variable.) Hence from any basic solution of DLP^k with nonvanishing denominator a basic solution of LFP^k can be obtained, too.

Furthermore, having a basis A_B of LFP^k with $\beta' \neq 0$, the equivalent basis of LP^k is

$$\text{(3.5)} \qquad B = \begin{bmatrix} A_B & -b \\ d_B^T & \beta \end{bmatrix}$$

and the inverse of B can be written as

$$\text{(3.6)} \qquad B^{-1} = \begin{bmatrix} A_B^{-1} + \dfrac{b' d_B^{T\prime}}{\beta'} & \dfrac{1}{\beta'} b' \\ \dfrac{d_B'^T}{\beta'} & \dfrac{1}{\beta'} \end{bmatrix},$$

where A_B, d_B are notations introduced in Section 2, $d'_B = -d_B^T A_B^{-1}$, $b' = A_B^{-1} b$ and $\beta' = \beta + d_B^T A_B^{-1} b$, i. e. β' is the current value of the denominator, and d'_B, b' are also obtainable knowing A_B^{-1} and the initial data of LFP^k.

Let an arbitrary basic solution of LFP^k be given in the form

$$\text{(3.7)} \qquad x_B^* = b' - A'_N x_N^*,$$

where $A'_N = A_B^{-1} A_N$ and x_B^* is the vector of the basic, x_N^* that of the nonbasic variables. As the simple lower and upper bounds are treated implicitly, x_{Nj}^* may stand for $x_{Nj} - L_{Nj}$, for $U_{Nj} - x_{Nj}$, or simply for x_{Nj}. Similarly, x_{Bi}^* means $x_{Bi} - L_{Bi}$, if $L_{Bi} \neq 0$.

The nominator and the denominator of the objective function can be written as

(3.8) $$f = \alpha' + c_N^{T\prime} x_N^*$$

and

(3.9) $$g = \beta' + d_N^{T\prime} x_N^*.$$

Suppose $\beta' \neq 0$. (The algorithm is organized in such a way that it should produce only basic solutions with $\beta' \neq 0$.)

The basic solution

(3.10) $$x_B^* = b', \; x_N^* = 0$$

is dual feasible, if

(3.11) $$\frac{1}{\beta'} \delta_N \leq 0,$$

where

(3.12) $$\delta_N = \beta' c_N' - \alpha' d_N'.$$

As for the primal feasibility, only those solutions whose equivalent is feasible for LP^k are considered primally feasible. For this reason two cases have to be distinguished:

a) If $\beta' > 0$, then (3.10) is primal feasible, if

(3.13) $$0 \leq b' \leq U_B^{k*},$$

where $U_B^{k*} = U_B^k - L_B^k$.

b) If $\beta' < 0$, (3.10) must not be feasible (see assumption A2). Here, unlike in LP, not only some x_{Bi}^*, but also x_{Nj}^* may have infeasible value (in the sense that the corresponding y variables in LP^k are infeasible). It can easily be seen from the corresponding basic solution of LP^k that x_{Bi}^* (i. e. the corresponding y_{Bi}^*) is infeasible, if

(3.14) $$\text{either } \frac{b_i'}{\beta'} < 0 \text{ or } \frac{U_{Bi}^{k*} - b_i'}{\beta'} < 0.$$

Furthermore, if x_{Nj} is an upper bounded variable with $U_{Nj}^{k*} > 0$, then either $x_{Nj}^* \left(\text{with } y_{Nj} = \frac{U_{Nj}^{k*}}{\beta'}\right)$ or $U_{Nj}^k - x_{Nj}^*$ $\left(\text{i. e. the slack variable of (2.10) with a value of } \frac{U_{Nj}^{k*}}{\beta'}\right)$ is infeasible.

Let the i-th row of (3.7) be written as either

$$x^*_{Bi} = b'_i - a^{T\prime}_i x^*_N \tag{3.15}$$

or

$$U^{k*}_{Bi} - x^*_{Bi} = U^{k*}_{Bi} - b'_i + a^{T\prime}_i x^*_N, \tag{3.16}$$

depending on which condition in (3.14) is fulfilled, and denote the j-th element of a'_i by a'_{ij}, and that of d'_N by d'_j.

The algorithm then can be formulated as follows.

Step 0: Start from a dual feasible basis with $\beta' \neq 0$.

Step 1 (Check for primal feasibility of basic variables):

See, if there exists i for which (3.14) is satisfied.
If yes, go to Step 2, otherwise go to Step 5.

Step 2: If in case of (3.15) either $\frac{1}{\beta'}a'_i \geq 0$ or $a'_i + \frac{b_i}{\beta'}d'_N \geq 0$, or in case of (3.16) either $\frac{1}{\beta'}a'_i \leq 0$ or $-a'_i + \frac{U^{k*}_{Bi} - b'_i}{\beta'} d'_N \geq 0$, then LFP_k has no feasible solution: go to Step 8.

Otherwise go to Step 3.

Step 3: Calculate

$$P_1 = \frac{\delta'_r}{a'_{ir}\beta' + b'_i d'_r} = \min_{j,\ \frac{a'_{ij}\beta' + b'_i d'_j}{\beta'} < 0} \left(\frac{\delta'_j}{a'_{ij}\beta' + b'_i d'_j} \right) \tag{3.17}$$

or

$$P_2 = \frac{\delta'_r}{-a'_{ir}\beta' + (U^{k*}_{Bi} - b'_i)d'_r} = \tag{3.18}$$

$$= \min_{j,\ \frac{-a'_{ij}\beta' + (U^{k*}_{Bi} - b'_i)d'_j}{\beta'} < 0} \left(\frac{\delta_j}{-a'_{ij}\beta' + (u^{k*}_{Bi} - b'_i)d'_j} \right)$$

Step 4 (Pivoting): *a)* If $a'_{ir} \neq 0$, then pivoting on a'_{ir} a new dual feasible basis of LFP^k is available in the form (3.7). Return to Step 1. *b)* If $a'_{ir} = 0$, the neighbouring basic solution of DLP^k does not correspond to any extremal point of X^k (the new A_B in 3.5 is singular). Go to Step 8.

Step 5 (Check for feasibility of the nonbasic variables): *a)* If $\beta' > 0$, the solution is optimal; go to Step 8. *b)* If $\beta' < 0$, see whether there exists a nonbasic x^*_{Nr} with upper bound $U^{k*}_{Nr} > 0$: if none, then either there does not exist a primal feasible solution with positive denominator, or the feasible region is unbounded: go to Step 8; if yes, go to Step 6.

Step 6: Calculate

$$P=\frac{\delta_q'}{\lambda_q\beta'+U_{Nr}^{k*}d_q'}=\min_{j,\,\lambda_j\beta'+U_{Nr}^{k*}d_j'>0}\left\{\frac{\delta_j'}{\lambda_j\beta'+U_{Nr}^{k}d_j'}\right\},$$

where $\lambda_j = 1$, if $j = r$, and $\lambda_j = 0$ otherwise.

Step 7 (Change of bounds): If there is no q in (3.19), then DLP^k is unbounded, i. e. LFP^k has no feasible solution with $\beta'>0$; go to Step 8.

Otherwise two cases have to be distinguished:

a) If $q = r$, exchanging the status of x_{Nr} (from lower bound to upper or vice versa) the basis remains dual feasible. Go to Step 1.

b) If $q \neq r$, the situation is similar to 4b; go to Step 8.

Step 8: End with either *a)* an optimal solution; or *b)* a dual feasible solution (the problem has to be solved by some other algorithm); or *c)* a conclusion that the problem is infeasible or the feasible region is unbounded.

Comments to the algorithm.

1. The validity of Step 2 follows from the duality theorems of linear programming.
2. The algorithm described above is essentially equivalent to the dual simplex algorithm of linear programming as applied to LP^k. On the basis of the connection between the corresponding basic inverses of LFP^k and LP^k (see (3.6)) any entry of the simplex tableau of LP^k can be obtained from the simplex tableau of LFP^k. For example, Steps 3 resp. 6 are minimal ratio calculations on a row of type (2.8) resp. (2.9) or (2.10) of LP^k. In case of Step 6 the corresponding row of LP^k is obtained from the implicit row $x_{Nr}^{*}\leq U_{Nr}^{k*}$ of LFP^k.
3. In Step 4a the new value of the objective function will be either

$$\frac{\alpha'}{\beta'}+\frac{b_i'}{\beta'}P_1\leq\frac{\alpha'}{\beta'} \tag{3.20}$$

or

$$\frac{\alpha'}{\beta'}+\frac{U_{Bi}^{k*}-b_i'}{\beta'}P_2\leq\frac{\alpha'}{\beta'}. \tag{3.21}$$

In Step 7a the new value of the objective function will be

$$\frac{\alpha'}{\beta'}+\frac{U_{Nr}^{k*}}{\beta'}P\leq\frac{\alpha'}{\beta'}. \tag{3.22}$$

4. In the singular cases of 4b or 7b the problem cannot be solved by this algorithm; we have to go on either with the dual simplex method on the extended tableau of LP^k until a basic solution with $t\neq 0$ is attained, and then return to Step 1 of this algorithm, or with Martos' primal method. Even in these cases, however, a bound on the value of the primal objective function can be obtained from (3.20), (3.21) or (3.22).

5. The notations like $x_j^* = U_j - x_j$ are used only for the sake of notational convenience; the actual algorithm works with the original variables with a book-keeping of their status.

4. Calculation of the penalties

Let us assume that the optimal solution of the k-th subproblem does not satisfy the integrality conditions, i. e. some basic variable x_{Bi}^* takes on a non-integer value in the basic solution (3.7)–(3.9). Then x_{Bi} can be written as

$$x_{Bi}^* = b_i' - a_i^{T\prime} x_N \tag{4.1}$$

with

$$b_i' = n_i + q_i, \tag{4.2}$$

where $n_i \geq 0$ is an integer number and $0 < q_i < 1$.

The immediate successors of LFP^k differ from LFP^k only in one of the bounds of the variable x_{Bi}:

$$X^{k+1} = X^k \cap \{x \in R^n \mid L_{Bi}^{k+1} = n_i + 1 \leq x_{Bi} \leq U_{Bi}^k\} \tag{4.3}$$

and

$$X^{k+2} = X^k \cap \{x \in R^n \mid L_{Bi} \leq x_{Bi} \leq n_i = U_{Bi}^{k+2}\}. \tag{4.4}$$

The optimal basic solution of DLP^k can be obtained from the optimal basis of LFP^k, which is equivalent to that of LP^k. Since x_{Bi}^* is basic in LFP^k, it follows that $w_{L_{Bi}} = w_{U_{Bi}} = 0$ in DLP^k; hence changing L_{Bi} or U_{Bi} does not influence the dual feasibility. (This is not obvious because of (3.3).) This means that the optimal solution of LFP^k is dually feasible for LFP^{k+1} and LFP^{k+2}.

Now it is easy to see that penalties, analogous to those proposed by Beale and Small [5] for MILP, can be derived for LFP^k (i. e. for LFP^{k+1} and LFP^{k+2}) on the basis of the dual algorithm described in Section 3:

$$PU = \min_j \{PU_j\} \tag{4.5}$$

$$PD = \min_j \{PD_j\} \tag{4.6}$$

where

$$PU_j = \begin{cases} \dfrac{1-q_i}{\beta'} \quad \dfrac{\delta_j'}{a_{ij}'\beta' + (q_i - 1)d_j'}, & \text{if } a_{ij}'\beta' + (q_i - 1)d_j' < 0 \\ \infty & \text{otherwise} \end{cases} \tag{4.7}$$

and

$$PD_j = \begin{cases} \dfrac{-q_i}{\beta'} \quad \dfrac{\delta_j'}{a_{ij}'\beta' + q_i d_j'}, & \text{if } a_{ij}'\beta' + p_i d_j' > 0 \\ \infty & \text{otherwise.} \end{cases} \tag{4.8}$$

Comparing (4.7) and (4.8) with (3.17) and (3.18), it can be observed that PU and PD are the decrease of the dual objective function resulting from implicitly carrying out one step of the dual algorithm on LFP^{k+1} and LFP^{k+2}, respectively. Hence

$$v^{k+1} \leq v^k - PU = z^k - PU \tag{4.9}$$

$$v^{k+2} \leq v^k - PU = z^k - PD \tag{4.10}$$

and since $z^{k+1} \leq v^{k+1}$ and $z^{k+2} \leq v^{k+2}$, an upper bound for all integer solutions of MILFP attainable from the current problem LFP^k is

$$z_*^k \leq z^k - \min\{PU, PD\}. \tag{4.11}$$

It can be shown that the penalties (4.5) and (4.6) can be improved in case of MILFP, too, in an analogous way as proposed by Tomlin in [15] for MILP.

The main point is, as in the linear case, that changing one of the bounds of x_{Bi}, at least one of the non-basic variables must be increased to have a feasible solution for LFP^{k+1} or LFP^{k+2}. Then we can make use of the fact that if some non-basic x_{Nj}^* is not to remain at zero, it must be at least one in any integer solution.

In the linear case the penalty for increasing a non-basic variable to one is simply its reduced cost, but in case of a fractional objective function the rate of change depends also on the value of the function itself, i. e. on the value of other non-basic variables. A similar, but not so easily computable penalty still can be given: the penalty for increasing x_{Nj}^* from 0 to 1 is

$$PI_j = \begin{cases} \min\limits_{r,\, \lambda_r \beta' + d'_r > 0} \left\{ \dfrac{-\delta_r}{\beta'(\lambda_r \beta' + d'_r)} \right\} \\ \infty, \text{ if no r with } \lambda_r \beta' + d'_r > 0 \text{ exists,} \end{cases} \tag{4.12}$$

where $\lambda_r = 1$ for $r = j$ and $\lambda_r = 0$ for $r \neq j$.

Note that (4.12) is analogous to (3.19). It can be obtained by implicitly carrying out a dual step on the (implicit) row $x_{Nj}^* \geq 1$.

Having PI_j the up and down penalties (4.5) and (4.6) can be replaced by the stronger ones:

$$PU^* = \min_j \begin{Bmatrix} PU_j & j \notin J \\ \max\{PI_j, PU_j\} & j \in J \end{Bmatrix} \tag{4.13}$$

$$PD^* = \min_j \begin{Bmatrix} PD_j & j \notin J \\ \max\{PI_j, PD_j\} & j \in J \end{Bmatrix}. \tag{4.14}$$

From the way PU and PD have been calculated it is straightforward that the overall penalty proposed by Tomlin [15] to be derived from the Gomory cut for any unsatisfied integer constraint (4.1)–(4.2) can be calculated, too. This penalty can be obtained by attaching implicitly the supplementary row of the Gomory constraint to LFP^k and carrying out, also implicitly, a dual step on the extended problem.

Thus, denoting the above-mentioned penalty by PG, the value z_*^k, of the best integer solution obtainable from LFP^k is bounded above by

(4.15) $$z_*^k \leq z^k - \max\{PG, \min\{PU^*, PD^*\}\}.$$

Similarly

(4.16) $$z_*^{k+1} \leq z^k - \max\{PU^*, PG\}$$

and

(4.17) $$z_*^{k+2} \leq z^k - \max\{PD^*, PG\}.$$

5. Concluding remarks

As it can be observed this paper has not outlined any actual B&B algorithm for MILFP, only the technique for solving the subproblems and the calculation of penalties have been described. The reason for this is that other aspects of solving MILFP do not differ from those of solving MILP. Actually any branching and backtracking strategy applicable for MILP can be applied for MILFP, too. Our purpose here was not to investigate the effect of different strategies on the efficiency of the algorithm, but only to show that the algorithm for MILFP is essentially equivalent to that for MILP.

The algorithm, without any modification, can be applied to solve as special cases MILP problems, too. The analogy is shown by the fact that using the same branching and backtracking strategy the same subproblems are generated, and the primal and dual simplex iterations are the same, too, as in the case of the MILP algorithm.

The MILFP algorithm with a simple version of branching and backtracking strategy has been implemented and tested on the R40 computer of the Computing Center of Eötvös Loránd University. The test problems were small MILFP problems and moderate size MILP problems.

6. Numerical example

Consider the problem

$$\text{maximize } \frac{2x_1+x_2-2}{x_1-x_2+1}$$

subject to

$$-5x_1+4x_2 \leq 0$$

$$-x_1+x_2 \leq 1/2$$

$$2x_1+x_2 \leq 11$$

$0 \leq x_1 \leq 5$, $0 \leq x_2 \leq 4$ required to be integers.

Introducing the slack variables x_3, x_4 and x_5 the simplex tableau containing the optimal solution of LFP° is

	x_4	$4-x_2$	
x_3	-5	1	$3/2$
x_5	2	-3	0
x_1	-1	1	$7/2$
$-f$	2	-3	-9
$-g$	1	0	$-1/2$
δ	-8	$-3/2$	

The value of the objective function corresponding to the solution $x_1 = 7/2$, $x_2 = 4$, $x_3 = 3/2$, $x_4 = x_5 = 0$ is $z^\circ = 18$ and $\beta' = 1/2 > 0$.

Since the value of x_1 is not integer, i. e. $q = 1/2 \neq 0$, two subproblems can be generated by replacing the bounds $0 \leq x_1 \leq 5$ by $4 \leq x_1 \leq 5$ for LFP¹ and by $0 \leq x_1 \leq 3$ for LFP².

The overall penalty is derived from the Gomory cut:

$$s = -1/2 - [-1,\ 0]\begin{bmatrix} x_4 \\ x_2^* \end{bmatrix} \geq 0.$$

The implicit dual step on this row results in

$$P_1 = \frac{-8}{-\frac{1}{2}-\frac{1}{2}} = 8 \text{ and } PG = \frac{-\frac{1}{2}}{-\frac{1}{2}} \cdot 8 = 8$$

(see (3.17) and (3.20)).

The penalties for LFP¹ resp. LFP² are calculated from the third row of the tableau by means of (4.7), i. e. (4.13) and (4.8) i. e. (4.14), respectively:

$$PU = \min\left\{\frac{1/2}{1/2} \cdot \frac{-8}{-1/2-1/2}, \infty\right\} = 8$$

$$PU^* = PU = 8$$

$$PD = \min\left\{\infty, \frac{-1/2}{1/2} \cdot \frac{-3/2}{1/2+0}\right\} = 3$$

$$PD^* = \min\{\infty, \max\{3,6\}\} = 6,$$

since the penalty for increasing $x_2^* = x_2 - 4$ from 0 to 1 is given by means of (4.12) as

$$PI_{x_3} = \min\left\{\frac{8}{\frac{1}{2}(0+1)}, \frac{3/2}{\frac{1}{2}\left(\frac{1}{2}+0\right)}\right\} = 6.$$

It follows that the upper bounds for the optimal solution are:

$$z_*^0 \leq 18-8 = 10, \quad z_*^1 \leq 18-8 = 10, \quad z_*^2 \leq 10.$$

A dual feasible basis of LFP^2 is available from that of LFP_0 by substituting $3-x_1$ for x_1 in the third row of the tableau. Then this row becomes

$3-x_1$	1	-1	$-1/2$

and pivoting on the element -1 as a result of Step 3 of the dual algorithm the optimal tableau of LFP^2 is

	x_4	$3-x_1$	
x_3	-4	1	1
x_5	-1	-3	$3/2$
x_2	1	1	$7/2$
$-f$	-1	-3	$-15/2$
$-g$	1	0	$-1/2$
δ	-8	$-3/2$	

Now x_2 is basic and not an integer. The subproblems of LFP^2 are generated by replacing the bounds $0 \leq x_2 \leq 4$ by $4 \leq x_2 \leq 4$ for LFP^3 and by $0 \leq x_2 \leq 3$ for LFP^4.

It is obvious that the constraint

$$x_2-4 = -\frac{1}{2}-[1, 1]\begin{bmatrix} x_4 \\ 3-x_1 \end{bmatrix} \geq 0$$

cannot be satisfied with $x_4 \geq 0$, $3-x_1 \geq 0$; therefore LFP^3 has no feasible solution and can be fathomed $(PU = \infty)$.

Calculating the penalties for the successors of LFP^2, $PG = 8$, $PD = 3$, $PD^* = 6$, the upper bound for any integer solution obtainable from LFP^4 is $z_*^4 \leq 15-8 = 7$.

A dual feasible basis of LFP^4 is obtained by substituting the row

$3-x_2$	-1	-1	$-1/2$

for the third row of the optimal tableau of LFP^2.

Then solving this problem by the dual method the optimal tableau of LFP^4 is

	x_4	$3-x_2$	
x_3	-5	1	$1/2$
x_5	2	-3	3
x_1	-1	1	$5/2$
$-f$	2	-3	-6
$-g$	1	0	$-1/2$
δ	-5	$-3/2$	

Here x_1 does not satisfy the integrality restriction, therefore the bounds $0 \leq x_1 \leq 3$ are replaced by $3 \leq x_1 \leq 3$ for LFP^5 and by $0 \leq x_1 \leq 2$ for LFP^6. The overall penalty for the successors of LFP^4 is $PG = 5$, and the penalties for the subproblems are $PU = PU^* = 5$ for LFP^5, $PD = 3$, $PD^* = 6$ for LFP^6.

The optimal solution of LFP^5 yields an integer solution $x_1 = x_2 = 3$, $x_3 = 3$, $x_4 = \frac{1}{2}$, $x_5 = 2$. The value of the objective function is $z_*^5 = 7$.

Considering now the pending subproblems LFP^1 and LFP^6, it can be seen that LFP^6 can be fathomed ($z_*^6 \leq z^4 - PD^* = 12-6 = 6$), but LFP^1 may yield potentially better integer solutions than the best one found so far.

Therefore x_1 in the third row of the optimal solution of LFP_0 has to be replaced by

x_1-4	-1	1	$-1/2$

As a result of Steps 3 and 4 of the dual algorithm the basic solution

	x_1-4	$4-x_2$	
x_3	-5	-4	4
x_5	2	-1	-1
x_4	-1	-1	$1/2$
$-f$	2	-1	-10
$-g$	1	1	-1
δ	-8	-11	

is obtained, which is still primally infeasible. Pivoting on the element -1 in the row of x_5, P_1 of (3.17) would be $P_1 = \frac{11}{2}$, therefore by means of (3.20) it is immediately seen that

$$z_*^1 \leq z^1 \leq v^1 \leq 10 + \frac{(-1)}{1} \cdot 11/2) = \frac{9}{2} < z_*^5 = 7,$$

that is LFP^1 can be fathomed and $z_* = 7$ is the optimal solution of the problem.

REFERENCES

[1] *Abrham J.* and *Luthra S.:* Comparison of duality models in fractional linear programming. *Zeitschrift für Operations Research* **21** (1977), 125–130.

[2] *Agrawal S. C.:* On integer solutions to linear fractional functionals by a branch and bound technique. *Acta Sciencia Indica* **2** (1976), 75–78.

[3] *Agrawal S. C.:* An alternative method on integer solutions to linear fractional functionals by a branch and bound technique. *Zeitschrift für Angewandte Mathematik und Mechanik* **57** (1977), 52–53.

[4] *Agrawal S. C.* and *Chand M.:* A note on integer solutions to linear fractional interval programming pıoblems by a branch and bound technique. *Naval Research Logistics Quarterly* **28** (1981), 671–677.

[5] *Beale E. M. L.:* Mathematical Programming in Practice. Pitman, London, 1968.

[6] *Beale E. M. L.* and *Small R. E.:* Mixed integer progıamming by a branch and bound technique. In: Pıoc. IFIP Congress 1965, Vol. 2. (ed. W. A. Kalenich). Spartan Press, Washington D. C. and Macmillan, London, 450–451.

[7] *Bitran G. R.* and *Magnanti T. L.:* Duality and sensitivity analysis for fractional programs. *Operations Research* **24** (1976), 675–699.

[8] *Chandra S.* and *Chandramohan M.:* An improved branch and bound method for mixed integer linear fractional programs. *Zeitschrift für Angewandte Mathematik und Mechanik* **59** (1979), 575–577.

[9] *Charnes A.* and *Cooper W.W.:* Programming with linear fractional functionals. *Naval Research Logistics Quarterly* **9** (1962), 181–186.

[10] *Land A. H.* and *Doig A. G.:* An automatic method of solving discrete programming problems. *Econometrica* **28** (1960), 497–520.

[11] *Land A. H.* and *Powell S.:* Computer codes for problems of integer programming. *Annals of Discrete Mathematics* **5** (1979), 221–269.

[12] *Martos B.:* Hyperbolic programming. *Naval Research Logistics Quarterly* **11** (1964), 135–155.

[13] *Schaible S.:* Fractional programming: applications and algorithms. *European Journal of Operational Research* **7** (1981), 111–120.

[14] *Schaible S.:* Bibliogıaphy in fractional programming. *Zeitschrift für Operations Research* **26** (1982), 211–241.

[15] *Tomlin J. A.:* Branch and bound methods for integer and non-convex programming. In: Integer and Non-Linear Programming (Ed. J. Abadie). North-Holland, Amsterdam, 1970, 437–450.

[16] *Wagner H. M.* and *Yuan J. S. C.:* Algorithmic equivalence in linear fractional programming. *Management Science* **14** (1968), 301–306.