An exact method for a discrete multiobjective linear fractional optimization

Chergui, M. E-A and Moulai, M.

LAID 3, Faculty of Mathematics, University of Sciences and Technology Houari Boumediene (USTHB), BP 32, Bab Ezzouar 16111, Algiers, Algeria, XLIM, institut de recherche, université de Limoges, France

9 June 2007

Hindawi Publishing Corporation
Journal of Applied Mathematics and Decision Sciences
Volume 2008, Article ID 760191, 12 pages
doi:10.1155/2008/760191

Research Article

An Exact Method for a Discrete Multiobjective Linear Fractional Optimization

Mohamed El-Amine Chergui and Mustapha Moulaï

LAID 3, Faculty of Mathematics, University of Sciences and Technology Houari Boumediene (USTHB), BP 32, Bab Ezzouar 16111, Algiers, Algeria

Correspondence should be addressed to Mohamed El-Amine Chergui, mchergui@usthb.dz

Received 9 June 2007; Revised 9 January 2008; Accepted 17 March 2008

Recommended by Wai-Ki Ching

Integer linear fractional programming problem with multiple objective (MOILFP) is an important field of research and has not received as much attention as did multiple objective linear fractional programming. In this work, we develop a branch and cut algorithm based on continuous fractional optimization, for generating the whole integer efficient solutions of the MOILFP problem. The basic idea of the computation phase of the algorithm is to optimize one of the fractional objective functions, then generate an integer feasible solution. Using the reduced gradients of the objective functions, an efficient cut is built and a part of the feasible domain not containing efficient solutions is truncated by adding this cut. A sample problem is solved using this algorithm, and the main practical advantages of the algorithm are indicated.

1. Introduction

Fractional programming has been widely reviewed by many authors (Schaible [1], Nagih, and Plateau [2]) and there are entire books and chapters devoted to this subject (Craven [3], Stancu-Minasian [4], Horst et al. [5], and Frenk and Schaible [6]). A bibliography, with 491 entries presented by Stancu-Minasian [7], attracts our attention to the amount of work that has been done in the field in recent years. This bibliography of fractional programming is a continuation of five previous bibliographies by the author [8]. Schaible [1] has published a comprehensive review of the work in fractional programming, outlining some of its major developments. Stancu-Minasian's textbook [4] contains the state-of-the-art theory and practice of fractional programming, allowing the reader to quickly become acquainted with what has been done in the field.

The mathematical optimization problems with a goal function that is a ratio of two linear functions have many applications: in finance (corporate planning, bank balance sheet management), marine transportation, water resources, health care, and so forth. Indeed, in such situations, it is often a question of optimizing a ratio debt/equity, output/employee, actual cost/standard cost, profit/cost, inventory/sales, risk-assets/capital, student/cost, doctor/patient, and so on subject to some constraints [9]. In addition, if the constraints are linear, we obtain the linear fractional programming (LFP) problem.

Different approaches have been proposed in the literature to solve both continuous LFP and integer linear fractional programming (ILFP) problems. These can be divided in studies that have developed solution methods (e.g., [4, 10–14]) and those which concentrated on applications (e.g., [4, 6]).

The multiple objective linear fractional programming (MOLFP) problem is one of the most popular models used in multiple criteria decision making. Numerous studies and applications have been reported in the literature in hundreds of books, monographs, articles, and books' chapters. For an overview of these studies and applications, see, for instance, [4, 7–9, 15–22], and references therein.

Contrary to the multiple objective linear programming (MOLP) problem, Steuer [9] shows that the efficient solutions set in MOLFP problem is not necessarily closed; some interior points of the feasible solutions set may be efficient, while others are not, and efficient extreme solutions need not all be connected by a path of efficient edges. It becomes difficult to generate the whole efficient solutions. As the efficient set may be too difficult to determine, Kornbluth and Steuer [20] propose an algorithm for MOLFP problem that generates the set of the so-called weakly efficient solutions by means of a simplex-based algorithm. A new technique to optimize a weighted sum of the linear fractional objective functions is proposed by Costa [19]. This technique generates only one nondominated solution of the MOLFP problem associated with a given weight vector. At each stage of the technique, the nondominated domain is divided in two subdomains and each of them is analyzed in order to discard the one not containing the nondominated solutions. The process ends when the remaining domains are so little that the differences among their nondominated solutions are lower than a predefined error.

In this paper, we have proposed a technique for generating the efficient set of the MOLFP problem with integer variables by using all the decision criteria in an active way. This last problem, called MOILFP, is more difficult to solve than the MOLFP problem taking into account the integrity of variables. Indeed, finding all efficient solutions of multiobjective combinatorial optimization problems is, in general, *NP-complete* [23].

We should like to point out that the MOILFP problem has not received as much attention as did the multiple objective integer linear programming (MOILP) problem, what justified our interest to study this problem.

In [15], a considerable computation is necessary to obtain an optimal integer solution of an ILFP problem in the first stage, since the authors used a branch and bound method (see, e.g., [24]). In our method, we use only the Cambini and Martein's [10] method to obtain an optimal solution for the relaxed ILFP problem and an integer solution is detected by the branching process of the branch and bound method. In addition, a cutting plane is constructed taking into account all the criteria. In this manner, we are able to eliminate not only noninteger solutions of the feasible domain, but also integer solutions which are not efficient. Thus our method avoids to scan all the integer feasible solutions of the problem.

The notations and definitions used throughout this work are presented in Section 2. In Section 3, the algorithm generating all efficient solutions for the MOILFP problem is developed and the main theoretical results are proposed. An illustrative example is given in Section 4 and a computational experience is reported in Section 5. Section 6 provides some concluding remarks.

2. Problem formulation

The purpose of this paper is to develop an exact method for solving the *multiple objective integer linear fractional program* (MOILFP):

$$(p)\begin{cases}\max Z_1(x) = \dfrac{c^1x+\alpha^1}{d^1x+\beta^1}\\ \max Z_2(x) = \dfrac{c^2x+\alpha^2}{d^2x+\beta^2}\\ \qquad\vdots\\ \max Z_k(x) = \dfrac{c^kx+\alpha^k}{d^kx+\beta^k}\\ x \in S, \qquad x\,\text{integer}\end{cases} \tag{2.1}$$

where $k \geq 2$; c^i, d^i are $1 \times n$ vectors; α^i, β^i are scalars for each $i \in \{1,2,\dots,k\}$; $S = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$; A is an $m \times n$ real matrix; and $b \in \mathbb{R}^m$. Throughout this article, we assume that S is a nonempty, compact polyhedron set in $\mathbb{R}^n$ and $d^ix+\beta^i > 0$ over S for all $i \in \{1,2,\dots,k\}$.

Many approaches for analyzing and solving the MOLFP problem use the concept of efficiency. A point $x \in \mathbb{R}^n$ is called an *efficient solution*, or *Pareto-optimal solution*, for MOLFP problem when $x \in S$, and there exists no point $y \in S$ such that $Z_i(y) \geq Z_i(x)$, for all $i \in \{1,\dots,k\}$ and $Z_i(y) > Z_i(x)$ for at least one $i \in \{1,\dots,k\}$. Otherwise, x is not efficient and the vector $Z(y)$ *dominates* the vector $Z(x)$, where $Z(x) = (Z_i(x))_{i=1,\dots,k}$.

The approach adopted in this work for detecting all integer efficient solutions of problem (P) is based on solving a linear fractional programming problem, at each stage l:

$$(p_1)\left\{\max Z_1(x) = \frac{c^1x+\alpha^1}{d^1x+\beta^1}, \quad x \in S_l,\right. \tag{2.2}$$

with $S_0 = S$ and without the integrity constraint of variables. Note that in place of Z_1, one can similarly consider the problem (P_l) with another objective Z_i for any $i \in \{2,\dots,r\}$.

If the optimal solution of (P_l) is integer, it is compared to all of the potentially efficient solutions already found and the set of efficient solutions is actualized. The growth direction of each criterion is determined by using its gradient. The method uses this information to deduce a cut able to delete integer solutions which are not efficient for the problem (P) and determines a new integer solution. In the case where this optimal solution is not integer, two new linear fractional programs are created by using the branching process well known in branch and bound method. Each of them will be solved like the problem (P_l).

To this aim—let x_l^* be the first integer solution obtained after solving problem (P_l) by using, eventually, the branching process—one defines B_l as the set of indices of basic variables and N_l as the set of indices of nonbasic variables of x_l^*. Let $\overline{\gamma_j^i}$ be the jth component of the reduced gradient vector $\overline{\gamma^i}$ defined by (2.3) for each fixed $i \in \{1,2,\dots k\}$;

$$\overline{\gamma^i} = \overline{\beta^i}\,\overline{c^i} - \overline{\alpha^i}\,\overline{d^i}, \tag{2.3}$$

where $\overline{c^i}$, $\overline{d^i}$, $\overline{\alpha^{i,}}$ and $\overline{\beta^i}$ are updated values. Let us note that the gradient vector of Z_i and the corresponding reduced gradient vector $\overline{\gamma^i}$ for each fixed index $i, i \in \{1, 2, \dots, k\}$, have the same sign. Thus calculating $\overline{\gamma^i}$ is enough to determine the growth direction for each criterion.

In order to give the mathematical expression of the cut, we define the following sets at x_l^*:

$$H_l = \left\{ j \in N_l \mid \exists i \in \{1, 2, \dots, k\};\ \overline{\gamma_j^i} > 0 \right\} \cup \left\{ j \in N_l \mid \overline{\gamma_j^i} = 0,\ \forall i \in \{1, 2, \dots, k\} \right\}, \tag{2.4}$$

$$S_{l+1} = \left\{ x \in S_l \mid \sum_{j \in H_l} x_j \geq 1 \right\}. \tag{2.5}$$

An *efficient cut* is a cut which removes only nonefficient integer solutions.

In Section 3, the approach to solve program (P) is presented.

3. Methodology for solving MOILFP

In this section, an exact method based on the branching process and using an efficient cut for generating all integer efficient solutions for problem (P) is presented. First of all, the proposed method is presented in detail, the algorithm for solving the multiple objective integer linear fractional programming problems is then described. We finish the section with the theoretical results which prove the convergence of the algorithm.

3.1. Description of the method

Starting with an optimal solution of an LFP problem, the domain of feasible integer solutions is partitioned into subdomains using the principle of branching to the search for integer solutions. As soon as an integer solution is found in a new domain, it is compared to solutions already found and hence the set of all the potentially efficient solutions is updated. An efficient cut is then added for deleting integer solutions that are not efficient. To construct this cut, the growth directions of the criteria are used. The search for the efficient solutions is made in each subdomain created. A given domain contains no efficient solutions when none criterion can grow. This last is said an explored domain. The search for the efficient solutions is stopped only if all created domains were explored domains.

First, Cambini and Martein's [10] algorithm is used for solving the following continuous linear fractional program:

$$(p) \begin{cases} \max Z_1(x) = \dfrac{c^1 x + \alpha^1}{d^1 x + \beta^1}, \quad x \in S_{0.} \end{cases} \tag{3.1}$$

This is based on the concept of optimal level solution. A feasible point $\overline{x}$ is an optimal level solution for the linear fractional program (P_0), if $\overline{x}$ is optimal for the linear program:

$$(P_{\overline{x}}) \begin{cases} \max \left(c^1 x + \alpha^1 \right), \quad d^1 x = d^1 \overline{x},\ x \in S_{0.} \end{cases} \tag{3.2}$$

If, in addition, $\overline{x}$ is a vertex of the feasible solutions set S_0, $\overline{x}$ is said to be a basic optimal level solution. Obviously, an optimal solution for the linear fractional program (P_0) is a basic optimal level solution. According to this, the algorithm generates a finite sequence of basic optimal level solutions, the first one, say x^0, is an optimal solution for the linear program:

$$\min\left(d^1 x + \beta^1\right), \quad x \in S_{0.} \tag{3.3}$$

If x^0 is unique, then it is also a basic optimal level solution for program (P_0), otherwise, solve the linear program (P_{x^0}) to obtain a basic optimal level solution.

The solution of the program (P_0) obtained in a finite sequence of optimal level solutions is optimal if and only if $\overline{\gamma_j^1} \leq 0$ for all $j \in J_0^1$, where

$$J_0^1 = \left\{ j \in N_0 \mid \overline{d_j^1} > 0 \right\}. \tag{3.4}$$

Otherwise, there exists an index $j \in J_0^1$ for which $\overline{\gamma_j^1} > 0$. The nonbasic variable $x_r, r \in J_0^1$, which must enters the basis is indicated by the index r such that

$$\frac{\overline{c_r^1}}{\overline{d_r^1}} = \max\left\{ \frac{\overline{c_j^1}}{\overline{d_j^1}}, \ j \in J_0^1 \right\}. \tag{3.5}$$

The original format of the objectives fractional functions and original structure of the constraints is maintained and the iterations are carried out in an augmented simplex table which includes $m + 3k$ rows. The first m rows correspond to the original constraints, the $m + 3(i-1) + 1$ and $m + 3(i-1) + 2$ rows correspond to the numerator and denominator of the objective fractional function $Z_i, i \in \{1, 2, \ldots, k\}$, of program (P), respectively, and the $m + 3i$ row corresponds to the $\overline{\gamma_l^i}$ vector at step l.

At each stage of the algorithm, all the rows are modified as usual through the pivot operation when the nonbasic variable $x_r, r \in J_0^1$, enters the basis, except the $m + 3i$ rows, for $i \in \{1, 2, \ldots, k\}$, which are modified using the $\overline{\gamma_l^i}$ formula (2.3).

Each program (P_l) corresponds to node l in a structured tree. A node l of the tree is fathomed if the corresponding program (P_l) is not feasible or $H_l = \varnothing$ (explored domain).

If the optimal solution $\tilde{x}_l$ of program (P_l) is not integer, let x_j be one component of $\tilde{x}_l$ such that $x_j = \alpha_{j,}$ where α_j is a fractional number. The node l of the tree is then separated in two nodes which are imposed by the additional constraints $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$, where $\lfloor \alpha_j \rfloor$ indicates the greatest integer less than α_j. In each node, the linear fractional program obtained must be solved, until an integer feasible solution is found. In presence of an integer feasible solution, the efficient cut $\sum_{j \in H_l} x_j \geq 1$ is added to the program and the new program is solved using the dual simplex method. The method terminates when all the created nodes are fathomed.

3.2. Algorithm

The algorithm generating the set of all integer efficient solutions of program (P) is presented in the following steps. The nodes in the tree structure are treated according to the backtracking principle.

Step 1. Initialization: $l = 0$, create the first node with the program (P_0). Eff $= \varnothing$; (integer-efficient set of program (P)

Step 2. General step: as long as a nonfathomed node exists in the tree, do: choose the node not yet fathomed, having the greatest number l, solve the corresponding linear fractional program (P_l) using the dual simplex method and the Cambini and Martein's method. (Initially, for solving program (P_0), only the Cambini and Martein's method is used).

If program (P_l) has no feasible solutions, then the corresponding node is fathomed.

Else, let $\tilde{x}_l$ be an optimal solution. If $\tilde{x}_l$ is not integer, go to Step 3, else go to Step 4.

Step 3. Branching process (partition of the problem into mutually disjoint and jointly exhaustive sub-problems): choose one coordinate x_j of $\tilde{x}_l$ such that $x_j := \alpha_j$, with α_j fractional number, and separate the actual node l of the tree in two nodes k, $k \geq l+1$, and h, $h \geq l+1$, $h \neq k$.

In the current simplex table, the constraint $x_j \geq \lfloor \alpha_j \rfloor$ is added and a new domain is considered in node k and similarly, the constraint $x_j \geq \lfloor \alpha_j \rfloor + 1$ is added to obtain another domain in node h. (Each created program must be solved using the same process until an integer feasible solution is found), go to Step 2.

Step 4. Update the set Eff: if $Z(\tilde{x}_l)$ is not dominated by $Z(x)$ for all $x \in \mathrm{Eff}$, then $\mathrm{Eff} := \mathrm{Eff} \cup \{\tilde{x}_l\}$. If there exists $x \in \mathrm{Eff}$ such that $Z(\tilde{x}_l)$ dominates $Z(x)$, then $\mathrm{Eff} := \mathrm{Eff} \setminus \{x\} \cup \{\tilde{x}_l\}$.

Construct the efficient cut: determine the sets N_l and H_l.

If $H_l = \emptyset$, then the corresponding node is fathomed. Go to Step 2.

Else, add the efficient cut $\sum_{j \in H_l} x_j \geq 1$ to the program (P_l). Go to Step 2.

The following theorems show that the algorithm generates all integer efficient solutions of program (P) in a finite number of stages.

Theorem 3.1. *Suppose that $H_l \neq \emptyset$ at the current integer solution x_l^*. If x is an integer efficient solution in domain $S_l \setminus \{x_l^*\}$, then $x \in S_{l+1}$.*

Proof. Let x be an integer solution in domain $S_l \setminus \{x_l^*\}$ such that $x \notin S_{l+1}$, then $\sum_{j \in H_l} x_j = 0$, that implies $x_j = 0$ for all index $j \in H_l$.

From the simplex table corresponding to the optimal solution x_l^*, the criteria are evaluated by

$$Z_i(x) = \frac{\sum_{j \in N_l} \overline{c_j^i} x_j + \overline{\alpha^i}}{\sum_{j \in N_l} \overline{d_j^i} x_j + \overline{\beta^i}} \quad \text{for } i \in \{1, \ldots, k\}, \tag{3.6}$$

where

$$\frac{\overline{\alpha^i}}{\overline{\beta^i}} = Z_i(x_l^*). \tag{3.7}$$

Then we can write

$$Z_i(x) = \frac{\sum_{j \in N_l \setminus H_l} \overline{c_j^i} x_j + \overline{\alpha^i}}{\sum_{j \in N_l \setminus H_l} \overline{d_j^i} x_j + \overline{\beta^i}}, \quad \forall i \in \{1, \ldots, k\}. \tag{3.8}$$

In the other hand, $\overline{\gamma_j^i} = \overline{\beta^i}\,\overline{c_j^i} - \overline{\alpha^i}\,\overline{d_j^i} \le 0$, for all index $j \in N_l \setminus H_l$ and $\overline{\gamma_j^i} = \overline{\beta^i}\,\overline{c_j^i} - \overline{\alpha^i}\,\overline{d_j^i} < 0$ for at least one criterion, implies that $\overline{c_j^i} \le \overline{\alpha^i}\,\overline{d_j^i}/\overline{\beta^i}$ for all $j \in N_l \setminus H_l$ because $\overline{\beta^i} = d^i x_l^* + \beta^i > 0$ for all criterion $i \in \{1, \dots, k\}$. The decision variables being nonnegative, we obtain $\overline{c_j^i} x_j \le (\overline{\alpha^i}\,\overline{d_j^i}/\overline{\beta^i}) x_j$ for all $j \in N_l \setminus H_l$ and hence

$$\sum_{j \in N_l \setminus H_l} \overline{c_j^i} x_j \le \sum_{j \in N_l \setminus H_l} \frac{\overline{\alpha^i}\,\overline{d_j^i}}{\overline{\beta^i}} x_j \Longrightarrow \sum_{j \in N_l \setminus H_l} \overline{c_j^i} x_j + \overline{\alpha^i} \le \sum_{j \in N_l \setminus H_l} \frac{\overline{\alpha^i}\,\overline{d_j^i}}{\overline{\beta^i}} x_j + \overline{\alpha^i}. \tag{3.9}$$

For any criterion Z_i, $i \in \{1, \dots, k\}$, the following inequality is obtained

$$\begin{aligned} Z_i(x) &= \frac{\sum_{j \in N_l \setminus H_l} \overline{c_j^i} x_j + \overline{\alpha^i}}{\sum_{j \in N_l \setminus H_l} \overline{d_j^i} x_j + \overline{\beta^i}} \\ &\Longrightarrow Z_i(x) \le \frac{\sum_{j \in N_l \setminus H_l} (\overline{\alpha^i}\,\overline{d_j^i}/\overline{\beta^i}) x_j + \overline{\alpha^i}}{\sum_{j \in N_l \setminus H_l} \overline{d_j^i} x_j + \overline{\beta^i}} \\ &\Longrightarrow Z_i(x) \le \frac{(\overline{\alpha^i}/\overline{\beta^i})\left(\sum_{j \in N_l \setminus H_l} \overline{d_j^i} x_j + \overline{\beta^i}\right)}{\sum_{j \in N_l \setminus H_l} \overline{d_j^i} x_j + \overline{\beta^i}} \\ &\Longrightarrow Z_i(x) \le \frac{\overline{\alpha^i}}{\overline{\beta^i}} \Longrightarrow Z_i(x) \le Z_i(x_l^*). \end{aligned} \tag{3.10}$$

Consequently, $Z_i(x) \le Z_i(x_l^*)$ for all $i \in \{1, \dots, k\}$ and $Z_i(x) < Z_i(x_l^*)$ for at least one index. Hence $Z(x_l^*)$ dominates $Z(x)$ and the solution x is not efficient. □

Corollary 3.2. *Suppose that $H_l \neq \varnothing$ at the current integer solution x_l^*. Then the constraint $\sum_{j \in H_l} x_j \ge 1$ is an efficient cut.*

Proof. By the above theorem, no efficient solution is deleted when the constraint $\sum_{j \in H_l} x_j \ge 1$ is added. We can say that this is an efficient valid constraint. In the other hand, x_l^* does not satisfy this constraint since $x_j = 0$, for all $j \in N_l$. We conclude that the constraint is an efficient cut. □

Proposition 3.3. *If $H_l = \varnothing$ at the current integer solution x_l^*, then $S_l \setminus \{x_l^*\}$ is an explored domain.*

Proof. $H_l = \varnothing$ means that x_l^* is an optimal integer solution for all criterion, hence x_l^* is an ideal point in the domain S_l and $S_l \setminus \{x_l^*\}$ does not contain efficient solutions. □

Theorem 3.4. *The described algorithm terminates in a finite number of iterations and generates all the efficient solutions of program (P).*

Proof. The set S of feasible solutions of problem (P), being compact, contains a finite number of integer solutions. Each time an optimal integer solution x_l^* is calculated, the efficient cut is added. Thus according to the above theorem and corollary, at least the solution x_l^* is eliminated when one studies any subproblem (P_k), $k > l$, but no efficient solution is deleted. □

Table 1

B	Rhs	x_3	x_4
x_2	$8/7$	$2/7$	$1/7$
x_1	$32/7$	$1/7$	$4/7$
c^1	$-4/7$	$-1/7$	$-4/7$
d^1	$-13/7$	$2/7$	$1/7$
γ^1		$-3/7$	$-8/7$
c^2	$4/7$	$1/7$	$4/7$
d^2	$-15/7$	$-2/7$	$-1/7$
γ^2		$1/7$	$8/7$
c^3	$24/7$	$-1/7$	$3/7$

4. An illustrative example

The following program (P) is given as an example of multiple objective linear fractional programming (MOLFP) in Kornbluth and Steuer [20]:

$$\begin{aligned}
&\max Z_1(x) = \frac{x_1 - 4}{-x_2 + 3}, \\
&\max Z_2(x) = \frac{-x_1 + 4}{x_2 + 1}, \\
&\max Z_3(x) = -x_1 + x_2, \quad \text{subject to} - x_1 + 4x_2 \leq 0,\ 2x_1 - x_2 \leq 8,\ x_1 \geq 0,\ x_2 \geq 0, \text{ and integers.}
\end{aligned} \tag{4.1}$$

Using the described algorithm, program (P_0) is first resolved and the optimal solution is given in the simplex Table 1:

Since $\gamma_j^1 \leq 0$ for all $j \in J_0^1$, $J_0^1 = N_0 = \{3,4\}$, then the obtained solution $(32/7, 8/7)$ is optimal for program (P_0), but not integer. Therefore, two branches are possible.

(1) $x_1 \geq 5 \Leftrightarrow -1/7x_3 - 4/7x_4 \geq 3/7$. This is not possible and (1) is fathomed.

(2) $x_1 \leq 4 \Leftrightarrow -1/7x_3 - 4/7x_4 \leq -4/7$.

This constraint is added and the dual simplex method is applied. The integer optimal solution of program (P_2) is obtained in Table 2.

$\gamma_j^1 \leq 0$ for all $j \in J_2^1$, $J_2^1 = N_2 = \{3,5\}$, then the current solution $(4,1)$ is optimal. Eff := $\{(4,1)\}$, $H_2 = \{5\}$, and $S_3 = \{x \in S_2 | x_5 \geq 1\}$.

The constraint $x_5 \geq 1$ is added to the current simplex table and after pivoting, we obtain Table 3.

$J_3^1 = \varnothing$ then, $(3,0)$ is an optimal integer solution and Eff := $\{(4,1),(3,0)\}$.

Proceeding in this manner, we obtain Table 4.

$J_4^1 = \varnothing$ then, $(0,0)$ is an optimal integer solution, Eff := $\{(4,1),(3,0),(0,0)\}$, $N_4 = \{1,10\}$, $H_4 = \{10\}$, and $S_5 = \{x \in S_4 | x_{10} \geq 1\}$.

The constraint $x_{10} \geq 1$ is added and we obtain Table 5.

The dual is not feasible, then the corresponding node is fathomed.

The algorithm terminates since all nodes are fathomed and the integer efficient solutions set of program (P) is Eff $= \{(4,1),(3,0),(2,0),(1,0),(0,0)\}$.

Table 2

B	Rhs	x_3	x_5
x_2	1	1/4	1/4
x_1	4	0	1
x_4	1	1/4	−7/4
c^1	0	0	−1
d^1	−2	1/4	1/4
γ^1		0	−2
c^2	0	0	1
d^2	−2	−1/4	−1/4
γ^2		0	2
c^3	3	−1/4	3/4

Table 3

B	Rhs	x_2	x_6
x_3	3	4	1
x_1	3	0	1
x_4	2	−1	−2
x_5	1	0	−1
c^1	1	0	−1
d^1	−3	−1	0
γ^1		−1	−3
c^2	−1	0	1
d^2	−1	1	0
γ^2		−1	1
c^3	3	1	1

Table 4

B	Rhs	x_1	x_{10}
x_6	3	1	0
x_3	0	1	4/3
x_4	8	1	−1
x_5	4	1	0
x_7	2	0	−1
x_9	1	3	8
x_8	0	1	1
x_2	0	−1	−1
c^1	4	1	0
d^1	−3	−1	−1
γ^1		−1	−4
c^2	−4	−1	0
d^2	−1	1	1
γ^2		−5	−4
c^3	0	0	1

Table 5

	Rhs	x_{11}	x_{10}
x_6	2	1	−1
x_3	−1	1	1/3
x_4	7	1	−2
x_5	3	1	−1
x_7	2	0	−1
x_9	−2	3	5
x_8	−1	1	0
x_2	1	−1	0
x_1	1	−1	1
c^1	3	1	−1
d^1	−2	−1	0
γ^1		−1	−4
c^2	−3	−1	1
d^2	−2	1	0
γ^2		−5	−4
c^3	0	0	1

Table 6

(n, m, α),		(15,10,33)	(20,10,25)	(25,5,17)	(25,10,17)
Efficient	Mean	99,50	204,50	200,45	98,00
Solutions	Max	215	324	397	228
	Min	4	23	67	17
CPU	Mean	38,52	185,96	400,48	306,74
(second)	Max	54,39	337,08	673,17	367,11
	Min	23,094	143,56	193,87	177,09
Simplex	Mean	75837,7	126488,50	521763,45	255726,55
Iterations	Max	101207	365750	719105	306665
	Min	55851	2717	369414	145324
Efficient	Mean	885,3	2341,50	2607,15	979,00
Cuts (EC)	Max	1152	2593	3909	1268
	Min	451	679	839	440
(EC)/(Cuts)	Mean	0,91	0,97	0,84	0,81

5. Computational results

The computer program was coded in MATLAB 7.0 and run on a 3.40 GHz DELL pentium 4, 1.00 GB RAM. The used software was developed by the authors and was tested on randomly generated problems. We show the results of the computational experiment in Table 6.

The method was tested with $m = 5$ and 10 constraints with $r = 4$ objectives functions and n variables, $n \in \{15, 20, 25\}$ randomly generated. The coefficients are uncorrelated integers uniformly distributed in the interval $[1, 100]$ for constraints and $[1, 80]$ for objective functions about the first two types of treated problems. For the latter two types of problems tested, the integer values of the matrix constraints vary in the interval $[1, 50]$ and for those of criteria in $[1, 30]$. The right-hand side value is set to $\alpha\%$ of the sum of the coefficients (integer part) of each constraint, where $\alpha \in \{17, 25, 33\}$. With each instance (n, m, α), a series of 20 problems is solved and the whole efficient solution set was generated for all these problems.

The method being exact, it was expected that the iteration number of the simplex method is very large taking into account the fact that, for this type of problems, the number of efficient solutions increases quickly with the data size. In addition, we should like to point out that the ratio EC/cuts tends toward the value one, showing that the number of efficient cuts introduced into the method is very large compared to the full number of cuts and indicating that this type of cuts has a positive impact on the research of the whole of efficient solutions.

6. Conclusion

In this paper, an exact method for generating all efficient solutions for multiple objective integer linear fractional programming problems is presented The method does not require any nonlinear optimization. A linear fractional program is solved using the Cambini and Martein's algorithm in the original format and then by using the well-known concept of branching in integer linear programming, integer solutions are generated. The proposed efficient cut exploits all the criteria in the simplex table, and only the parts of the feasible solutions domain containing efficient solutions are explored. Also it is easy to implement the proposed cut since to obtain integer solution x_{k+1} from x_k, one has just to append the cut in the simplex table corresponding to x_k and carry out pivoting iterations as in an ordinary linear fractional programming problem. The described method solves MOILFP problems in the general case. However, in order to make the algorithm more powerful, the tree structure of the algorithm can be exploited for construction of a parallel algorithm. For large scale problems, the number of efficient solutions can be very high so that it becomes unrealistic to generate them all. In this case, one can choose only the increasing directions of criteria which satisfy a desirable augmentation. This can be made by building the sets H_l in an interactive way at each step of the algorithm.

Acknowledgments

The authors are grateful to anonymous referees for their substantive comments that improved the content and presentation of the paper. This research has been completely supported by the laboratory LAID3 of the High Education Algerian Ministry.

References

[1] S. Schaible, "Fractional programming: applications and algorithms," *European Journal of Operational Research*, vol. 7, no. 2, pp. 111–120, 1981.

[2] A. Nagih and G. Plateau, "Problèmes fractionnaires: tour d'horizon sur les applications et méthodes de résolution," *RAIRO Operations Research*, vol. 33, no. 4, pp. 383–419, 1999.

[3] B. D. Craven, *Fractional Programming*, vol. 4 of *Sigma Series in Applied Mathematics*, Heldermann, Berlin, Germany, 1988.

[4] I. M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*, vol. 409 of *Mathematics and Its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[5] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*, vol. 48 of *Nonconvex Optimization and Its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2nd edition, 2000.

[6] J. B. G. Frenk and S. Schaible, "Fractional programming: Introduction and Applications," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds., pp. 162–172, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[7] I. M. Stancu-Minasian, "A sixth bibliography of fractional programming," *Optimization*, vol. 55, no. 4, pp. 405–428, 2006.

[8] I. M. Stancu-Minasian, "A fifth bibliography of fractional programming," *Optimization*, vol. 45, no. 1–4, pp. 343–367, 1999.

[9] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley & Sons, New York, NY, USA, 1986.

[10] A. Cambini and L. Martein, "Equivalence in linear fractional programming," *Optimization*, vol. 23, no. 1, pp. 41–51, 1992.

[11] A. Charnes and W. W. Cooper, "Programming with linear fractional functionals," *Naval Research Logistics Quarterly*, vol. 9, no. 3-4, pp. 181–186, 1962.

[12] D. Granot and F. Granot, "On integer and mixed integer fractional programming problems," in *Studies in Integer Programming*, vol. 1 of *Annals of Discrete Mathematics*, pp. 221–231, North-Holland, Amsterdam, The Netherlands, 1977.

[13] B. Martos, *Nonlinear Programming: Theory and Method*, North-Holland, Amsterdam, The Netherlands, 1975.

[14] C. R. Seshan and V. G. Tikekar, "Algorithms for integer fractional programming," *Journal of the Indian Institute of Science*, vol. 62, no. 2, pp. 9–16, 1980.

[15] M. Abbas and M. Moulaï, "Integer linear fractional programming with multiple objective," *Journal of the Italian Operations Research Society*, vol. 32, no. 103-104, pp. 15–38, 2002.

[16] R. Caballero and M. Hernández, "The controlled estimation method in the multiobjective linear fractional problem," *Computers & Operations Research*, vol. 31, no. 11, pp. 1821–1832, 2004.

[17] A. Cambini, L. Martein, and I. M. Stancu-Minasian, "A survey of bicriteria fractional problems," *Advanced Modeling and Optimization*, vol. 1, no. 1, pp. 9–46, 1999.

[18] M. Chakraborty and S. Gupta, "Fuzzy mathematical programming for multi objective linear fractional programming problem," *Fuzzy Sets and Systems*, vol. 125, no. 3, pp. 335–342, 2002.

[19] J. P. Costa, "Computing non-dominated solutions in MOLFP," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1464–1475, 2007.

[20] J. S. H. Kornbluth and R. E. Steuer, "Multiple objective linear fractional programming," *Management Science*, vol. 27, no. 9, pp. 1024–1039, 1981.

[21] B. Metev and D. Gueorguieva, "A simple method for obtaining weakly efficient points in multiobjective linear fractional programming problems," *European Journal of Operational Research*, vol. 126, no. 2, pp. 386–390, 2000.

[22] O. M. Saad and J. B. Hughes, "Bicriterion integer linear fractional programs with parameters in the objective functions," *Journal of Information & Optimization Sciences*, vol. 19, no. 1, pp. 97–108, 1998.

[23] M. Ehrgott and X. Gandibleux, "An annotated bibliography of multiobjective combinatorial optimization," Report in Wissenschaftmathematik no. 62, Fachbereich Mathematik, Universitat Kaiserslautern, Kaiserslautern, Germany, 2000.

[24] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, NY, USA, 1986.