AN EXACT METHOD FOR OPTIMIZING TWO LINEAR FRACTIONAL
FUNCTIONS OVER THE EFFICIENT SET OF A MULTIOBJECTIVE
INTEGER LINEAR FRACTIONAL PROGRAM

Abstract
In this paper, an exact method is proposed to optimize two fractional linear functions over the efficient set of a fractional multiobjective linear problem (MOILFP). This type of problems is encountered when there are two decision makers and each has his own utility function that he wants to optimize over the efficient set of multiobjective problem. The proposed method uses Branch and Bound method combined with a cutting plane technique to find the efficient solutions for both utility functions and (MOILFP) without going through all the efficient solutions of the two problems. An illustrative example and a computational study are reported.

Keywords Multiobjective programming · Fractional programming · Integer programming · Branch-and-cut · Nonlinear programming

1 Introduction
Optimizing a nonlinear or linear function over the efficient set is an interesting area of multiobjective programming (Miettinen, 1999). It is a simple way to avoid listing all efficient solutions by evaluating and distinguishing them from each other using a function that summarizes the preferences of decision makers.

First considered by Philip (1972), the problem of optimization over the efficient set has since attracted the attention of several authors, among whom Jorge (2009) who proposed an exact method that successively solves single-objective programs, Zerdani and Moulaï (2011) have optimized a linear function over the integer efficient set of MOILFP using Ehrgott’s efficiency test (1997). More recently, Drici et al. (2018) also proposed an exact method that uses the well-known concept of branch-and-bound combined with efficient cuts technique and Liu and Ergoth (2018) have presented primal and dual algorithms to solve this problem. Other entries can be found in Yamamoto’s survey (2002).

Most of the time, in this type of problems, we have several decision makers and each one of them can have a utility function. Cherfaoui and Moulaï (2019) have treated the case where there are two linear functions to optimize over the efficient set of an integer linear multiobjective problem. However, in practice, measuring a quality, a profitability, a probability,..., are described as fractional functions. Fractional programming is encountered in several areas of application such as : stock cutting problem (Gilmore & Gomory, 1961), shape-quality optimization (Munson, 2007), clustering problems (Rao, 1971), etc., more applications of fractional programming can be found in Schaible (1981), Stancu-Minasian’s bibliography (2013).

In this article, we generalize the latter method (Cherfaoui & Moulaï, 2019) where we have two linear fractional functions (BOILFP) that we want to optimize over the efficient set of (MOILFP). This problem was encountered in a real world problem summarized as follow: A company "A" wants to subcontract a part of its production. On another hand, a Company "B" wants to take over the project and at the same time.
has several fractional linear objective functions. The company "B" wants to know if there is one or more solutions to benefit both companies. In other words, finding the solutions that are efficient for both the (BIOLFP) problem and the (MOILFP) problem. One way to solve this problem is to enumerate the efficient sets of both (BIOLFP) and (MOILFP) and find the intersection between the two efficient sets. This method has an important computational cost as we will show it in the computational study.

The proposed method is an exact branch and cut procedure that provides the exact efficient solutions for both (BIOLFP) and (MOILFP). This procedure has the advantage to avoid useless exploration of domain by using cuts that eliminates dominated solutions, pruning by the same way useless nodes in the tree. It also uses efficiency tests that allow only the good solutions to be selected.

2 Definitions and preliminaries

Linear Fractional Programs (LFP) are very important because of their contribution to real problems (Bajalinov, 2013; Stancu, 2012), their general formulation is:

\[
(LFP) \begin{cases}
    \max f_1(x) = \frac{p^1 x + \alpha^1}{q^1 x + \beta^1} \\
    \text{s.t.} \\
    x \in \mathcal{X}
\end{cases}
\]  

(1)

where \( \mathcal{X} = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\} \), \( p^1, q^1 \) are \( n \)-vectors, \( \alpha^1 \) and \( \beta^1 \) are scalars, \( b \) is an \( m \)-vector, \( A \) is an \( n \times m \)-matrix and \( x \) is the \( n \)-vector decision.

The set \( \mathcal{X} \) is assumed to be nonempty convex polytope and the set \( \mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n \) is assumed not empty as well.

Several authors proposed approaches that can solve (LFP) efficiently. In the first place, Charnes and Cooper (1962) proposed a variable transformation that turns the problem into a linear program. Cambini and Martein (1986) proposed a simplex procedure improved from the procedure of Martos (1964). This procedure uses the reduced gradient to move towards the optimum for a base \( B_l \) and associated solution \( x^{(l)} \) using the following notations:

\[
\begin{align*}
    \nu^{(l)} &= p^1 - p^1_{B_l} B^{-1}_{l} A \\
    \mu^{(l)} &= q^1 - q^1_{B_l} B^{-1}_{l} A, \\
    P^1(x) &= p^1 x^{(l)} + \alpha^1 \\
    Q^1(x) &= q^1 x^{(l)} + \beta^1
\end{align*}
\]

the reduced gradient for the problem (1) can be formulated as:

\[
\gamma^{(l)} = Q^1(x)\nu^{(l)} - P^1(x)\mu^{(l)}
\]

The reduced gradient gives the direction of growth of the objective function \( f_1(x) \), so if for all index \( j \) from the non-basic index set \( N_l \), \( \gamma_j^{(l)} \leq 0 \) then the function \( f_1(x) \) cannot increase.

Most of real life problems can be modelled as multiobjective problems with integer numbers, therefore let us consider the following multiobjective program:

\[
(MOILFP) \begin{cases}
    \max Z_1(x) = \frac{c^1 x + c^1_0}{d^1 x + d^1_0} \\
    \max Z_2(x) = \frac{c^2 x + c^2_0}{d^2 x + d^2_0} \\
    \quad \vdots \\
    \max Z_k(x) = \frac{c^k x + c^k_0}{d^k x + d^k_0} \\
    \text{s.t.} \\
    x \in \mathcal{D}
\end{cases}
\]  

(2)
where $k \geq 2$; $c^i, d^i$ are $n-$vectors; $c^i_0, d^i_0$ are scalars for each $i \in \{1, 2, \ldots, k\}$. Throughout this article, we assume that $d^i x + d^i_0 > 0$ over $X$ for all $i \in \{1, 2, \ldots, k\}$.

**Definition 2.1.** A solution $x \in D$ is called an efficient solution for $(MOILFP)$, if there exists no point $y \in D$ such that $Z_i(y) \geq Z_i(x)$, for all $i \in \{1, \ldots, k\}$ and $Z_i(y) > Z_i(x)$ for at least one $i \in \{1, \ldots, k\}$.

Some researchers tackled this problem in their papers especially in the continuous case; Kornbluth and Steuer (1981) presented a simplex-based solution procedure to find all weakly efficient vertices, while Costa (2007) proposed a new technique to optimize a weighted sum of the linear fractional objective functions. Finally, Cambini et al. (1999) wrote a survey on the biobjective fractional problems.

In general, since it is computationally heavy, authors avoid trying to generate the whole efficient set. However, Chergui and Moulaï (2008) used a branch-and-cut procedure to generate the whole efficient set. In their paper, they combined branch-and-bound exploration with a cutting plane technique to generate the efficient set without enumerating the whole integer domain.

In this paper, we propose a method that generates a part of the efficient set of $(MOILFP)$ that is defined by decision makers preferences. Therefore, let us assume the existence of two utility functions $f_1$ and $f_2$:

\[
\begin{align*}
(BOILFP) \quad \max f_1(x) &= \frac{p^1 x + \alpha^1}{q^1 x + \beta^1} \\
\max f_2(x) &= \frac{p^2 x + \alpha^2}{q^2 x + \beta^2} \\
\text{s.t.} & \quad x \in D
\end{align*}
\]

where $p^1, p^2, q^1, q^2$ are $n-$vectors; $\alpha^1, \beta^1$ and $\alpha^2, \beta^2$ are scalars. Throughout this article, we also assume that $q^1 x + \beta^1 > 0$ and $q^2 x + \beta^2 > 0$ over $X$ and the following notations are used for $B_i, N_i$ and $x^{s(i)}$:

\[
\begin{align*}
P^s(x) &= p^s x + \alpha^s, \\
z^s_1(x) &= c^s x + c^s_0, \\
Q^s(x) &= q^s x + \beta^s, \\
z^s_2(x) &= d^s x + d^s_0, \\
\nu^{s(i)} &= p^s - p^s_i B_i^{-1} A, \\
\mu^{s(i)} &= q^s - q^s_i B_i^{-1} A, \\
\eta^{s(i)} &= c^s - c^s_i B_i^{-1} A, \\
\phi^{s(i)} &= d^s - d^s_i B_i^{-1} A.
\end{align*}
\]

Therefore, the problem that we propose to solve is:

\[
(B_E) \quad \max f_1(x) = \frac{p^1 x + \alpha^1}{q^1 x + \beta^1} = \frac{p^1 x + \alpha^1}{q^1 x + \beta^1} 
\]

\[
\text{s.t.} \quad x \in X_E,
\]

where $X_E$ is the set of efficient solutions of $(MOILFP)$, we also note $X_E$: the set of efficient solutions of the $(BOILFP)$. The problem $(B_E)$ has a real practical utility, yet it has not been considered. It arises, for example, whenever two firms have to optimize their respective interest and a common problem.

### 2.1 Efficiency test for multi-objective linear fractional programs

The efficiency test of integer feasible solutions $x^*$ of the problem $(2)$ is verified by solving the mixed integer linear program $(5)$.

\[
MM(x^*) \quad \max \psi = \sum_{i=1}^{k} \psi_i 
\]

\[
\text{s.t.} \quad \begin{align*}
&[c^i - Z_i(x^*)] d^i x - \psi_i = Z_i(x^*) d^i_0 - c^i_0, \quad i = 1, \ldots, k \quad x \in D; \\
&\psi_i \geq 0, \text{ for all } i \in \{1, \ldots, k\}
\end{align*}
\]
where \( Z_i(x^*) = \frac{d_i^j x^* + c_i^j}{d_i^* x^* + d_0^j} \) and \( \psi = (\psi_i)_{i \in \{1, \ldots, K\} \in \mathbb{R}}. \)

**Theorem 2.2.** A feasible solution \( x^* \) of problem (2) is efficient if and only if the optimal objective value of problem \( M M(x^*) \) is zero.

**Proof.** Let \((y, \psi)\) be any feasible solution of problem \( M M(x^*) \), then for all \( i = 1, \ldots, k \), we have:

\[
\psi_i \geq 0, \\
\iff (c_i^j - Z_i(x^*) d_i^j)y - Z_i(x^*) d_0^j + \lambda c_i^j \geq 0, \\
\iff Z_i(y) \geq Z_i(x^*).
\]

Assume that \( \Psi = \sum_{i=1}^{k} \psi_i \neq 0 \), then there exists \( r \in \{1, \ldots, k\} \) such that

\[
\psi_r > 0 \iff (c_r^j - Z_r(x^*) d_r^j)y - Z_r(x^*) d_0^j + c_r^j > 0 \\
\iff Z_r(y) > Z_r(x^*),
\]

therefore, there exists a feasible solution \( y \in D \) such that \( Z_i(y) \geq Z_i(x^*) \) for all \( i \in \{1, \ldots, k\} \) and \( Z_r(y) > Z_r(x^*) \) for at least one \( r \in \{1, \ldots, k\} \). Hence, \( x^* \) is not efficient.

Conversely, let \((y, \psi)\) be an optimal solution of \( M M(x^*) \), where \( \Psi = 0 \), then we have

\[
\sum_{i=1}^{k} \psi_i = 0 \implies \psi_i = 0, \quad i = 1, \ldots, k, \text{ it follows that for all criteria } (Z_i)i = 1, \ldots, k
\]

\[
(c_i^j - Z_i(x^*) d_i^j)y - Z_i(x^*) d_0^j + c_i^j = 0, \\
\iff c_i^j y + c_i^j = Z_i(x^*)(d_i^j y + d_0^j), \\
\iff Z_i(y) = Z_i(x^*).
\]

Hence, the criterion vector \((Z_1(x^*), Z_2(x^*), \ldots, Z_k(x^*))\) is not dominated, then \( x^* \) is efficient. \( \square \)

### 3 Methodology, Algorithm and Theoretical Results

The naive approach to solve \((B_E)\) consists in enumerating the elements of \( \mathcal{X}_E \) and \( \mathcal{X}_E' \) and determine their intersection. In what follows, we propose a method that avoids the enumeration of both and gives the exact solution set of \((B_E)\). This method is a branch-and-cut procedure that combines the branch-and-bound exploration technique and cutting plane method. The solution set of \((B_E)\) is noted \((S_E)\).

In this paper, we propose to use an equivalent cutting plane method of Chergui and Moulaï (2008) to remove solutions that do not belong to \( \mathcal{X}_E \) or \( \mathcal{X}_E' \). Our method can be summarized as follows:

#### The branching process

The search for efficient solutions is an implicit exploration structured as a tree where \( f_1 \) or \( f_2 \) can be chosen to be optimized at each node. The choice of function \( f_1 \) or \( f_2 \) will not change the result set \( S_E \), however, it will alter the search tree and change the evolution of \( S_E \) but not the final result.

At each node \( l \) of the tree we optimize the function \( f_l \) over the subdomain corresponding to the node \( \mathcal{X}_l. \)

\[
\{LFP\} \left\{ \begin{array}{ll}
\max f_l(x) = \frac{p^l x + \alpha^l}{q^l x + \beta^l} \\
\text{s.t.} \\
x \in \mathcal{X}_l
\end{array} \right.
\]

where \( \mathcal{X}_0 = \mathcal{X} \) and \( \mathcal{X}_l \) is subset of the original set \( \mathcal{X} \) to explore at node \( l \) of the tree.

The functions \( f_1 \) and \( f_2 \) are linear fractional. To solve the \((LFP_l)\), we can either use the method presented in (Charnes & Cooper, 1962) or another simplex method like those presented in (Cambini & Martein, 1986, Martos, 1964).

After solving \((LFP_l)\), three cases can occur:

- **(LFP_l) has no solution** As for a classical Branch-and-Bound process the node has no descendant.

- **(LFP_l) has a non-integer solution \( x^*(l) \)** If the solution of \((LFP_l)\) is not integer, the approach is the same as the Branch-and-Bound for integer programming, which means that the node has two descendants \( l_1 \) and \( l_2 \) such as: Let \( r \) be an index where \( x^{*(l)}_r \) is not integer then for \( l_1 \) we put \( \mathcal{X}_l_1 = \{x \in \mathcal{X}_l | x_r \leq \lfloor x^{*(l)}_r \rfloor \}\) and for \( l_2 \) we put \( \mathcal{X}_l_2 = \{x \in \mathcal{X}_l | x_r \geq \lceil x^{*(l)}_r \rceil \}\).
(LPF) has an integer solution $x^{*(l)}$ If after calculating the solution $x^{*(l)}$, this solution is integer then two steps are performed:

- **The first step is to test the membership of $x^{*(l)}$ to $S_E$:***
  
  Since there are two efficiency tests to take into account:

  $$(T^1_{x^{*(l)}}) \begin{cases} 
  \max \sum_{i=1}^{k} w_i \\
  \text{s.t.} \\
  (c_i - Z_i(x^{*(l)}))d_i - w_i = Z_i(x^{*(l)})d_0 - e_i, \ i = 1, k \\
  x \in D \\
  \max v_1 + v_2 
  \end{cases}
  $$

  $$(T^2_{x^{*(l)}}) \begin{cases} 
  \max v_1 + v_2 \\
  \text{s.t.} \\
  (p^1 - f_1(x^{*(l)})q^1)x - v_1 = f_1(x^{*(l)})\beta^1 - \alpha^1 \\
  (p^2 - f_2(x^{*(l)})q^2)x - v_2 = f_2(x^{*(l)})\beta^2 - \alpha^2 \\
  x \in D 
  \end{cases}
  $$

  The first program is for testing the efficiency in the program (MOILFP) and the second for the program (BIOILFP). If both programs have a zero as an optimal value it means that the solution $x^{*(l)}$ is efficient for (MOILFP) and (BIOILFP) and thus $x^{*(l)} \in S_E$.

- **The second is to search the descendants of the node l:** The node in this case has either no descendant or one direct descendant depending on the growth direction of the functions $Z_i, i \in \{1, \ldots, k\}$ and $f_2$. If there is no increasing for all the functions $Z_i$ in the domain $X_i$ then $x^{*(l)}$ dominates all the domain $X_i$ in terms of (MOILFP) and the node is fathomed. Similarly if there is no solution that is at least better than $x^{*(l)}$ in the criterion $f_2$, then the solution $x^{*(l)}$ dominates the whole domain $X_i$ in terms of (BOILFP) and the node is fathomed.

  On the other hand, if there is a possible improvement of one of the functions $Z_i, i \in \{1, \ldots, k\}$ and $f_2$, an efficient solution in terms of (MOILFP) and (BOILFP) might still exists in $X_i$ and the node has descendants. So, the cutting plane method that we use permit to remove solutions that are dominated by $x^{*(l)}$ criteria vectors.

  Let $B_i$ and $N_i$ be the sets of basic variables and non-basic variables indexes of $x^{*(l)}$ respectively. In what follows we will use these notations for $j \in N_i$:

  $$
  \gamma^{2(l)} = Q^2(x^{*(l)})\nu^{2(l)} - P^2(x^{*(l)})\mu^{2(l)} \\
  \lambda^{i(l)} = z^2(x^{*(l)})\eta^{i(l)} - z^1(x^{*(l)})\theta^{i(l)}, \ i \in \{1, 2, \ldots, k\}
  $$

  Since $\gamma^{1(l)}$ and $\gamma^{2(l)}$ represent the reduced gradient vectors for $f_1$ and $f_2$, and $\lambda^{i(l)}$ corresponds to the reduced gradient vectors for the criterion $Z_i$ for each $i \in \{1, 2, \ldots, k\}$. We define the sets $H_l$ and $H'_l$ for each node $l$ as follows:

  $$
  H_l = \{ j \in N_i | \exists i \in \{1, \ldots, k\}; \lambda^{i(l)}_j > 0 \} \cup \{ j \in N_i | \gamma^{j(l)} = 0, \forall i \in \{1, \ldots, k\} \}
  $$

  $$
  H'_l = \{ j \in N_i | \gamma^{j(l)} > 0 \} \cup \{ j \in N_i | \gamma^{j(l)} = 0 \text{ and } \gamma^{j(l)} = 0 \}
  $$

  In the following theorem [4.1] we will prove that the cuts:

  $$
  \sum_{j \in H_l} x_j \geq 1 \text{ and } \sum_{j \in H'_l} x_j \geq 1
  $$

  are valid and eliminate solutions that are dominated by $x^{*(l)}$.

  After finding an integer solution, we update $S_E$ and calculate $H_l, H'_l$. If $H_l = \emptyset$ or $H'_l = \emptyset$ it means that the remaining domain contains no efficient solution either in terms of (MOILFP) or (BOILFP), and the node $l$ is fathomed (See proposition [4.2]). Otherwise, if $H_l \neq \emptyset$ and $H'_l \neq \emptyset$ the node $l$ has one successor $l_0$ with $X_{l_0} = \{ x \in X_l | \sum_{j \in H_l} x_j \geq 1, \sum_{j \in H'_l} x_j \geq 1 \}$.

  Thus, there exists two rules of fathoming nodes, the first rule is when the corresponding program (LPF) is not feasible and the second is when $H_l = \emptyset$ or $H'_l = \emptyset$.
In the following algorithm, the nodes in the tree structure can be treated according to the backtracking or the depth first principles or any other exploration strategy.

**Algorithm 1: Biobjective optimization over MultiObjective Integer Linear Fractional Program**

**Result:** The set \( S_E \) containing all the efficient solutions of the program (4) initialization \( l = 0, \mathcal{X}_l = \{ x \in \mathbb{R}^n | Ax \leq b \text{ and } x \geq 0 \} \) and \( S_E = \emptyset \); **while there is a non-fathomed node \( l \) do**

solve \((LFP_l)\); \[
\begin{align*}
(LFP_l) \quad \max & \quad f_1(x) \\
\text{s.t.} & \quad x \in \mathcal{X}_l
\end{align*}
\]

if \((LFP_l)\) has an optimal solution \( x^*(l) \) then

if \( x^*(l) \) is integer then

Update \( S_E \);

Construct the sets \( \mathcal{H}_l, \mathcal{H}'_l \);

if \( \mathcal{H}_l = \emptyset \) or \( \mathcal{H}'_l = \emptyset \) then

Fathom the node \( l \);

else

Add the cuts \((6)\) and \((7)\) to \( l_0 \) the successor of \( l \);

end

else

Choose an index \( r \) such as \( x^*_r(l) \) is fractional. Then, split the program \((LFP_l)\) into two sub-programs, by adding respectively the constraints \( x_r \leq \lfloor x^*_r(l) \rfloor \) and \( x_r \geq \lceil x^*_r(l) \rceil \) to obtain \((LFP_{l_1})\) and \((LFP_{l_2})\) (\( l_1 \geq l + 1, l_2 > l + 1 \) and \( l_1 \neq l_2 \));

end

else

Fathom the node \( l \);

end

end

4 **Theoretical results**

The following theoretical tools show that the algorithm yields the solution set \( S_E \) of the program (4) \((B_E)\) in a finite number of iterations.

**Theorem 4.1.** Assume that \( \mathcal{H}_l \neq \emptyset \) and \( \mathcal{H}'_l \neq \emptyset \) at the current integer solution \( x^*(l) \). If \( x \in S_E, x \neq x^*(l) \) and \( x \in \mathcal{X}_l, then x \in \mathcal{X}_{l_1} \) \((l_1 \text{ successor of } l)\).

**Proof.** Let \( x \neq x^*(l) \) be an integer solution in domain \( \mathcal{X}_l \) such as \( x \notin \mathcal{X}_l \), two cases can occur:

- \( x \notin \mathcal{X}_l \) because \( x \in \{ x \in \mathcal{X}_l \mid \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} x_j \geq 1 \} \), this implies \( x \in \{ x \in \mathcal{X}_l \mid \sum_{j \in \mathcal{H}_l} x_j < 1 \} \).

Therefore, the following inequalities hold:

\[
\sum_{j \in \mathcal{H}_l} x_j < 1 \quad \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} x_j \geq 1.
\]

It follows that \( x_j = 0 \) for all \( j \in \mathcal{H}_l \) and \( x_j \geq 1 \) for at least one index \( j \in \mathcal{N}_l \setminus \mathcal{H}_l \).
From the optimal simplex table corresponding to solution \( x^{*(l)} \), the updated value of each objective function is written according to the non-basic indexes \( j \in \mathcal{N}_l \) as follows:
\[
\begin{align*}
  z_1^i(x) &= z_1^i(x^{*(l)}) + \delta_j \eta_j^{i(l)}, \\
  z_2^i(x) &= z_2^i(x^{*(l)}) + \delta_j \varphi_j^{i(l)},
\end{align*}
\]
where \( \delta_j = \frac{x_{B_i(x^*)}^j}{A_j^*} = \min \left\{ \frac{x_{B_i(x^*)}^j}{A_j^*} \mid A_j^* > 0 \right\} \).

Thus, we have for all \( i \in \{1, 2, \ldots, k\} \):
\[
Z_i(x) = \frac{z_1^i(x)}{z_2^i(x)} = \frac{z_1^i(x^{*(l)}) + \delta_j \eta_j^{i(l)}}{z_2^i(x^{*(l)}) + \delta_j \varphi_j^{i(l)}},
\]
then we can write
\[
Z_i(x) - Z_i(x^{*(l)}) = \frac{z_1^i(x^{*(l)}) + \delta_j \eta_j^{i(l)}}{z_2^i(x^{*(l)}) + \delta_j \varphi_j^{i(l)}} - \frac{z_1^i(x^{*(l)})}{z_2^i(x^{*(l)})}
\]
\[
= \delta_j \left[ \frac{z_2^i(x^{*(l)}) \eta_j^{i(l)} - z_1^i(x^{*(l)}) \varphi_j^{i(l)}}{z_2^i(x^{*(l)}) + \delta_j \varphi_j^{i(l)}} \right]
\]
\[
= \delta_j \left[ \frac{z_2^i(x^{*(l)}) \eta_j^{i(l)} - z_1^i(x^{*(l)}) \varphi_j^{i(l)}}{z_2^i(x^{*(l)} \times z_2^i(x^{*(l)})} \right]
\]

Since we have already the following notation:
\[
\lambda^{i(l)} = z^2(x^{*(l)}) \eta_j^{i(l)} - z^1(x^{*(l)}) \varphi_j^{i(l)}, \quad i = 1, \ldots, k
\]
As the components \( \lambda^{i(l)} \leq 0 \), for every index \( j \in \mathcal{N}_l \setminus \mathcal{H}_l \)
\[
Z_i(x) - Z_i(x^{*(l)}) \leq 0, \quad i = 1, \ldots, k
\]
and for at least one criterion \( i_0 \in \{1, \ldots, r\} \), \( Z_{i_0}(x) - Z_{i_0}(x^{*(l)}) < 0 \).
Hence, the criterion vector \( (Z_1(x), Z_2(x), \ldots, Z_k(x)) \) is dominated by the criterion vector \( (Z_1(x^{*(l)}), Z_2(x), \ldots, Z_k(x^{*(l)})) \) then \( x \notin \mathcal{X}_E \). Hence \( x \notin \mathcal{S}_E \)

\( \bullet \) \( x \notin \mathcal{X}_l \) because \( x \in \left\{ x \in \mathcal{X}_l \mid \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} x_j \geq 1 \right\} \), by the same way it implies that
\[
x \in \left\{ x \in \mathcal{X}_l \mid \sum_{j \in \mathcal{H}_l} x_j < 1 \right\}.
\]

Therefore, the following inequalities hold:
\[
\sum_{j \in \mathcal{H}_l} x_j < 1 \quad \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} x_j \geq 1.
\]
It follows that \( x_j = 0 \) for all \( j \in \mathcal{H}_l \) and \( x_j \geq 1 \) for at least one index \( j \in \mathcal{N}_l \setminus \mathcal{H}_l \).

From the optimal simplex table corresponding to solution \( x^{*(l)} \), the updated value of each objective function is written according to the non-basic indexes \( j \in \mathcal{N}_l \) as follows:
\[
P^2(x) = P^2(x^{*(l)}) + \delta_j \mu_j^{2(l)},
\]
\[
Q^2(x) = Q^2(x^{*(l)}) + \delta_j \mu_j^{2(l)}.
\]
The algorithm terminates in a finite number of iterations and the set $S_E$ contains all solutions of $B_E$, if such solutions exist.

Proof. Let $D$, the set of integer feasible solutions of $MOILFP$ problem, be a finite bounded set contained in $X$. The cardinality of efficient sets $S_E$ and $X_E$, $X_E$ is a finite number. It contains a finite number of integer solutions. So the search tree would have a finite number of branches. Thus the algorithm terminates in a finite number of steps.

Now, assume that $\mathcal{H}_i = \emptyset$, then $\forall j \in N_i$, $\forall x \in D_i$, we have $\overline{\lambda}_j \leq 0$ and $\exists \lambda_0 \in \{1, \ldots, r\}$ such that $\overline{\lambda}_j \leq 0$, $\forall j \in N_i$. So, $x^{*(l)}$ dominates all points $x, x \neq x^{*(l)}$ of domain $D_i$.

Now, assume that $\mathcal{H}_i = \emptyset$, then $\forall j \in N_i$, $\overline{\lambda}_j < 0$ or $\overline{\lambda}_j = 0$ and $\overline{\lambda}_j < 0$, adding to that $\overline{\lambda}_j < 0$, $\forall j \in N_i$ since it is an optimal solution for $(LP_i)$, $x^{*(l)}$ becomes the most preferred solution in the domain $D_i$.

Theorem 4.3. The algorithm terminates in a finite number of iterations and the set $S_E$ contains all solutions of $B_E$, if such solutions exist.

Proof. Let $D$, the set of integer feasible solutions of $MOILFP$ problem, be a finite bounded set contained in $X$. The cardinality of efficient sets $S_E$ and $X_E$, $X_E$ is a finite number. It contains a finite number of integer solutions. So the search tree would have a finite number of branches. Thus the algorithm terminates in a finite number of steps.

For $S_E$ to contain all the solutions of $(B_E)$, the fathoming rules are used without loss of any elements in $S_E$. At each step $l$ of the algorithm [1] if an integer solution $x^{*(l)}$ is found, the cuts eliminate $x^{*(l)}$ and all dominated solutions from search (see proposition 4.2).
So, the first fathoming rule is when the set $H_l$ or $H'_l$ is empty. In this case, the current node can be pruned since the rest of the domain contains only dominated solutions, either in terms of the multiobjective program or in terms of $(BIOLFP)$. The second rule is the trivial case when the reduced domain becomes infeasible, whether it is because of previous cuts or the branching. 

5 Illustrative example

Let us consider the optimization of two utility functions:

$$f_1(x) = \frac{-x_1 + x_2 - 3}{2x_1 + x_1 + 1} \quad \text{and} \quad f_2(x) = \frac{-4x_1 + 3x_2 + 1}{2x_1 + x_2 + 2}$$

over the efficient set $\mathcal{X}_E$ of the $(MOILFP)$ presented in (Kornbluth & Steuer, [1981]):

$$\begin{align*}
\max & \quad x_1 - 4 \\
\max & \quad -x_2 + 2 \\
\max & \quad -x_1 + 4 \\
\max & \quad x_2 + 1 \\
\max & \quad -x_1 + x_2 \\
\text{s.t.} & \quad -x_1 + 4x_2 \leq 0 \\
& \quad 2x_1 - x_2 \leq 8 \\
& \quad x_1 \geq 0, \ x_2 \geq 0 \text{ and integers.}
\end{align*}$$

To solve this problem using the described algorithm, the steps below will be followed:

Initialization Firstly we initialise $l = 0$, $S_E = \emptyset$ and $\mathcal{X}_0 = \mathcal{X}$.

the forward iterations consist in solving the problem attached to the node, so for the node 0 we will solve $(LFP_0)$:

$$\begin{align*}
(LFP_0) \quad \max & \quad f_1(x) = \frac{-x_1 + x_2 - 3}{2x_1 + x_1 + 1} \\
\text{s.t.} & \quad x \in \mathcal{X}_0
\end{align*}$$

Since the method of solution of $(LFP_0)$ is a simplex procedure it gives a final simplex table, from which we will analyse the results according to the described algorithm.

Node 0 solving $(LFP_0)$ gives the final simplex Table [1]:

9
Table 1: Optimal simplex table for node 0.

|       | $x_3$ | $x_4$ | RHS  |
|-------|-------|-------|------|
| $x_1$ | 1/7   | 4/7   | 32/7 |
| $x_2$ | 2/7   | 1/7   | 8/7  |
| $\nu^{1(0)}$ | -1/7  | 3/7   | -45/7 |
| $\mu^{1(0)}$ | -4/7  | -9/7  | 79/7  |
| $\gamma^{1(0)}$ | -37/7 | -24/7 | -45/79 |
| $\nu^{2(0)}$ | -2/7  | 13/7  | -97/7 |
| $\mu^{2(0)}$ | -4/7  | -9/7  | 86/7  |
| $\gamma^{2(0)}$ | -80/7 | 5      | -97/86 |
| $\nu^{3(0)}$ | -1/7  | -4/7  | 4/7  |
| $\mu^{3(0)}$ | 2/7   | 1/7   | 6/7  |
| $\gamma^{3(0)}$ | -2/7  | -4/7  | 2/3  |
| $\nu^{4(0)}$ | 1/7   | 4/7   | -4/7 |
| $\mu^{4(0)}$ | -2/7  | -1/7  | 15/7 |
| $\gamma^{4(0)}$ | 1/7   | 8/7   | -4/15 |
| $\nu^{5(0)}$ | -1/7  | 3/7   | -24/7 |
| $\mu^{5(0)}$ | 0     | 0     | 1    |
| $\gamma^{5(0)}$ | -1/7  | 3/7   | -24/7 |

Since the optimal solution $\left(\frac{32}{7}, \frac{8}{7}\right)$ is not integer, the problem is divided into two sub-problems by adding constraints $x_1 \leq \bigg\lfloor \frac{32}{7} \bigg\rfloor$ and $x_1 \geq \bigg\lceil \frac{32}{7} \bigg\rceil$ respectively to $X_0$, and we obtain the two programs:

\[
\begin{align*}
(LFP_1) & \quad \max f_1(x) = -\frac{x_1 + x_2 - 3}{2x_1 + x_1 + 1} \\
& \text{s.t.} \\
& x \in X_0 \\
& x_1 \leq 4
\end{align*}
\]

\[
\begin{align*}
(LFP_2) & \quad \max f_1(x) = -\frac{x_1 + x_2 - 3}{2x_1 + x_1 + 1} \\
& \text{s.t.} \\
& x \in X_0 \\
& x_1 \geq 5
\end{align*}
\]

Node 1 Solving the problem $(LFP_1)$ gives the following Table 2.
Table 2: Optimal simplex table for node 1.

| $B_1$ | $x_3$ | $x_5$ | RHS |
|-------|-------|-------|-----|
| $x_1$ | 0     | 1     | 4   |
| $x_2$ | 1/4   | 1/4   | 1   |
| $x_4$ | 1/4   | -7/4  | 1   |

| | $\nu^{1(1)}$ | $\mu^{1(1)}$ | $\gamma^{1(1)}$ |
| | -1/4 | -1/4 | -4 |
| | 3/4  | -9/4 | -6  |
| | -6   | -3/5 |

| | $\nu^{2(1)}$ | $\mu^{2(1)}$ | $\gamma^{2(1)}$ |
| | -3/4 | 1/4 | -1/4 |
| | 13/4 | 1/4 | -1/4 |
| | -12  | 11  |

| | $\nu^{2(1)}$ | $\mu^{2(1)}$ | $\gamma^{2(1)}$ |
| | -3/4 | 1/4 | -1/4 |
| | 13/4 | 1/4 | -1/4 |
| | -12  | 11  |

| | $\eta^{1(1)}$ | $\omega^{1(1)}$ | $\lambda^{1(1)}$ |
| | 0     | 1     | 0   |
| | 1/4   | 1/4   | 1   |
| | 0     | -1    | 0   |

| | $\eta^{2(1)}$ | $\omega^{2(1)}$ | $\lambda^{2(1)}$ |
| | 0     | 1     | 0   |
| | 1     | -1/4  | 2   |
| | 0     | 2     | 0   |

| | $\eta^{3(1)}$ | $\omega^{3(1)}$ | $\lambda^{3(1)}$ |
| | -1/4 | 3/4 | -3 |
| | 0     | 0    | 1   |
| | -1/4 | 3/4 | -3 |

The optimal solution found $(4, 1)$ is integer, according to the algorithm we start by updating $S_E$, since this set is empty $S_E = \{(4, 1)\}$ then we calculate $H_1$ and $H'_1$: From the Table 2 we find $H_1 = \{5\}$ and $H'_1 = \{5\}$. So the Node 0 has one successor Node 3 with $\lambda_3 = \lambda_1 \cap \{x_5 \geq 1\}$.

**Node 2** The program $(LFP_2)$ is infeasible.

**Node 3** After solving the corresponding linear fractional program, it gives Table 3.
Table 3: Optimal simplex table for node 3.

| $B_3$ | $x_3$ | $x_6$ | RHS |
|-------|-------|-------|-----|
| $x_1$ | 0     | 1     | 3   |
| $x_2$ | 1/4   | 1/4   | 3/4 |
| $x_4$ | 1/4   | -7/4  | 11/4|
| $x_5$ | 0     | -1    | 1   |
| $\nu^{1(3)}$ | -1/4 | 3/4  | -21/4 |
| $\mu^{1(3)}$ | -1/4 | -9/4 | 31/4  |
| $\gamma^{1(3)}$ | -13/4 | -6   | -21/31 |
| $\nu^{2(3)}$ | -3/4 | 13/4 | -35/4 |
| $\mu^{2(3)}$ | -1/4 | -9/4 | 35/4  |
| $\gamma^{2(3)}$ | -35/4 | 35/4 | -1 |
| $\eta^{1(3)}$ | 0     | -1    | -1  |
| $\vartheta^{1(3)}$ | 1/4   | 1/4   | 5/4  |
| $\lambda^{1(3)}$ | 1/4   | -1    | -4/5 |
| $\eta^{2(3)}$ | 0     | 1     | 1   |
| $\vartheta^{2(3)}$ | -1/4  | -1/4  | 7/4  |
| $\lambda^{2(3)}$ | 1/4   | 2     | 4/7  |
| $\nu^{3(3)}$ | -1/4 | 3/4  | -9/4 |
| $\vartheta^{3(3)}$ | 0     | 0     | 1   |
| $\lambda^{3(3)}$ | -1/4 | 3/4  | -9/4 |

The optimal solution $(3, \frac{3}{4})$ is not integer so the constraints $x_2 \leq \lfloor \frac{3}{4} \rfloor$ and $x_1 \geq \lceil \frac{3}{4} \rceil$ are added respectively to Table 3 and the nodes 4 and 5 are created.

**Node 4** Solving the problem gives Table 4.
Table 4: Optimal simplex table for node 4.

| $B_4$ | $x_6$ | $x_7$ | RHS |
|-------|-------|-------|-----|
| $x_1$ | 0     | 1     | 3   |
| $x_2$ | 1     | 0     | 0   |
| $x_3$ | -4    | -1    | 3   |
| $x_4$ | 1     | -2    | 2   |
| $x_5$ | 0     | -1    | 1   |
| $\nu^{1(3)}$ | -1 | 1     | -6  |
| $\mu^{1(3)}$ | -1 | -2    | 7   |
| $\gamma^{1(3)}$ | -13 | -5   | -6/7 |
| $\nu^{2(3)}$ | -3  | 4     | -11 |
| $\mu^{2(3)}$ | -1  | -2    | 8   |
| $\gamma^{2(3)}$ | -35 | 10    | -11/8 |
| $\eta^{1(3)}$ | 0   | -1    | -1  |
| $\theta^{1(3)}$ | 1   | 0     | 2   |
| $\lambda^{1(3)}$ | 1   | -2    | -1/2 |
| $\eta^{2(3)}$ | 0   | 1     | 1   |
| $\theta^{2(3)}$ | -1  | 0     | 1   |
| $\lambda^{2(3)}$ | 1   | 1     | 1   |
| $\nu^{3(3)}$ | -1  | 1     | -3  |
| $\theta^{3(3)}$ | 0   | 0     | 1   |
| $\lambda^{3(3)}$ | -1  | 1     | -3  |

The optimal solution found $(3, 0)$ is integer, this solution is dominated by the solution $(4, 1)$ then $S_E = \{(4, 1)\}$. From the Table we find $H_4 = \{6, 7\}$ and $H_4' = \{6\}$, so the Node 4 has one successor Node 6 with $X_6 = X_4 \cap \{x_6 + x_7 \geq 1, x_6 \geq 1\}$.

Node 5 The corresponding problem is infeasible and the node is fathomed.

Node 6 By adding the constraints $x_6 \geq 1$ and $x_6 + x_7 \geq 1$ to Table and solving, we obtain Table.
Table 5: Optimal simplex table for node 6.

| $B_6$ | $x_7$ | $x_9$ | RHS |
|-------|-------|-------|-----|
| $x_1$ | 0     | 1     | 2   |
| $x_2$ | 1     | 0     | 0   |
| $x_3$ | -4    | 1     | 2   |
| $x_4$ | 1     | -2    | 4   |
| $x_5$ | 0     | -1    | 2   |
| $x_6$ | -0    | -1    | 1   |
| $x_8$ | -1    | -1    | 0   |

| $\nu^{1(6)}$ | -1 | 1 | -5 |
| $\mu^{1(6)}$ | -1 | -2 | 5 |
| $\gamma^{1(6)}$ | -10 | -5 | -1 |

| $\nu^{2(6)}$ | -3 | 4 | -7 |
| $\mu^{2(6)}$ | -1 | -2 | 6 |
| $\gamma^{2(6)}$ | -25 | 10 | -7/6 |

| $\eta^{1(6)}$ | 0 | -1 | -2 |
| $\vartheta^{1(6)}$ | 1 | 0 | 2 |
| $\lambda^{1(6)}$ | 2 | -2 | -1 |

| $\eta^{2(6)}$ | 0 | 1 | 2 |
| $\vartheta^{2(6)}$ | -1 | 0 | 1 |
| $\lambda^{2(6)}$ | 2 | 1 | 2 |

| $\eta^{3(6)}$ | -1 | 1 | -2 |
| $\vartheta^{3(6)}$ | 0 | 0 | 1 |
| $\lambda^{3(6)}$ | -1 | 1 | -2 |

The optimal solution found $(2, 0)$ is integer, this solution is dominated by the solution $(4, 1)$ then $S_E = \{(4, 1)\}$. From the Table 5 we find $H_6 = \{7, 9\}$ and $H'_6 = \{9\}$, so the Node 6 has one successor Node 7 with $X_7 = X_6 \cap \{x_7 + x_9 \geq 1, x_9 \geq 1\}$.

**Node 7** Solving the corresponding problem produces Table [6](#)
Table 6: Optimal simplex table for node 7.

| $E_7$ | $x_7$ | $x_{10}$ | RHS |
|-------|-------|--------|-----|
| $x_1$ | 0     | 1      | 1   |
| $x_2$ | 1     | 0      | 0   |
| $x_3$ | -4    | 1      | 1   |
| $x_4$ | 1     | -2     | 6   |
| $x_5$ | 0     | -1     | 3   |
| $x_6$ | 0     | -18    | 2   |
| $x_8$ | 0     | -1     | 1   |
| $x_9$ | -1    | -1     | 1   |
| $x_{11}$ | -1 | -1 | 0 |

| $\nu^1(7)$ | -1 | 1 | -4 |
| $\mu^1(7)$ | -1 | -2 | 3 |
| $\gamma^1(7)$ | -7 | -5 | -4/3 |

| $\nu^2(7)$ | -3 | 4 | -3 |
| $\mu^2(7)$ | 1 | -2 | 4 |
| $\gamma^2(7)$ | -15 | 10 | -3/4 |

| $\eta^1(7)$ | 0 | -1 | -3 |
| $\phi^1(7)$ | 1 | 0 | 2 |
| $\lambda^1(7)$ | 3 | -2 | -3/2 |

| $\eta^2(7)$ | 0 | 1 | 3 |
| $\phi^2(7)$ | -1 | 0 | 1 |
| $\lambda^2(7)$ | 3 | 1 | 3 |

| $\eta^3(7)$ | -1 | 1 | -1 |
| $\phi^3(7)$ | 0 | 0 | 1 |
| $\lambda^3(7)$ | -1 | 1 | -1 |

The optimal solution found (1, 0) is integer, this solution is not dominated by the solution (4, 1) so $S_E = \{(4, 1), (1, 0)\}$. From the Table we find $H_7 = \{7, 10\}$ and $H'_7 = \{10\}$, so the Node 7 has one successor Node 8 with $X_8 = X_7 \cap \{x_7 + x_{10} \geq 1, x_{10} \geq 1\}$. Node 8 Solving the corresponding problem produces Table.
The optimal solution found \((0, 0)\) is integer, this solution is not dominated by the either \((4, 1)\) or \((0, 0)\), it doesn’t dominate both so \(\mathcal{S}_E = \{(4, 1), (1, 0), (0, 0)\}\). From the Table 7 we find \(\mathcal{H}_8 = \{7, 13\}\) and \(\mathcal{H}_0 = \{13\}\), so the Node 8 has one successor Node 9 with \(\mathcal{X}_9 = \mathcal{X}_8 \cap \{x_7 + x_{10} \geq 1, x_{13} \geq 1\}\)

### Node 9
The corresponding problem is infeasible and the node is fathomed.

There remains no nodes so the search is finished with the solution set \(\mathcal{S}_E = \{(4, 1), (1, 0), (0, 0)\}\). The efficient set of this problem is \(\mathcal{X}_E = \{(4, 1), (3, 0), (2, 0), (1, 0), (0, 0)\}\)

The search tree is presented in Figure[1]

---

**Table 7: Optimal simplex table for node 8**

| \(B_8\) | \(x_7\) | \(x_{12}\) | RHS |
|--------|--------|--------|-----|
| \(x_1\) | 0      | 1      | 0   |
| \(x_2\) | 1      | 0      | 0   |
| \(x_3\) | -4     | 1      | 0   |
| \(x_4\) | 1      | -2     | 8   |
| \(x_5\) | 0      | -1     | 4   |
| \(x_6\) | 0      | -1     | 3   |
| \(x_8\) | 0      | -1     | 2   |
| \(x_9\) | -1     | -1     | 2   |
| \(x_{10}\) | 0      | -1     | 1   |
| \(x_{11}\) | -1     | -1     | 1   |
| \(x_{12}\) | -1     | -1     | 0   |

\(\nu^{1(8)}\): -1 1 -3

\(\mu^{1(8)}\): -1 -2 1

\(\gamma^{1(8)}\): -4 -5 -3

\(\nu^{2(8)}\): -3 4 1

\(\mu^{2(8)}\): -1 -2 2

\(\gamma^{2(8)}\): -5 10 1/2

\(\eta^{1(8)}\): 0 -1 -4

\(\theta^{1(8)}\): 1 0 2

\(\lambda^{1(8)}\): -4 -2 -2

\(\eta^{2(8)}\): 0 1 4

\(\theta^{2(8)}\): -1 0 1

\(\lambda^{2(8)}\): 4 1 4

\(\nu^{3(8)}\): -1 1 0

\(\mu^{3(8)}\): 0 0 1

\(\lambda^{3(8)}\): -1 1 0
6 Computational results

The proposed branch-and-cut algorithm is implemented in Matlab r2017a. The linear and integer linear programs are solved using the library IBM CPLEX 12.9 for Matlab. To perform tests we have used a computer with an Intel i7 5500u processor and 8GB of memory.

The method is tested on randomly generated (MOILFP) with two randomly generated utility linear fractional functions. The objective functions and constraints coefficients are uncorrelated uniformly distributed. Each component of the vector $b$ and the entries of the matrices $A$ were randomly drawn from discrete uniform distributions in the ranges $[50, 100]$, $[1, 30]$ respectively. The vectors $p^1$, $p^2$ and $c^i$, $i = 1, k$ and the scalars $\alpha^1$, $\alpha^2$ and $c^i_0$ are randomly drawn from discrete uniform distributions in the range $[-10, 10]$, while the vectors $q^1$, $q^2$ and $d^i$, $i = 1, k$ and the scalars $\beta^1$, $\beta^2$ and $d^i_0$ are generated randomly from discrete uniform distributions in the range $[0, 10]$ to avoid the indefinite case of the null denominator. Also, to avoid infeasibility, all the constraints of each problem are of the $\leq$ type. Furthermore, since all coefficients of $A$ are positive, a bounded feasible region is assured.

The problems were grouped according to the number of variables, constraints and objective functions into six categories. In each category the number of objective functions $r = 3, 5, 7$. For each category of problems, 10 instances were solved.

The computational results obtained are summarized in Table 8. The statistics of the $CPU$ time (in seconds) and the number of nodes are reported. The last column $\mu$ refers to the average of $|\mathcal{X}_E|$, while $n$ and $m$ refer to the number of variables and constraints respectively. The cases where the execution time required to find the efficient set $\mathcal{X}_E$ is too high are noted $-\infty$.
Table 8: Random instances execution time

| $r \times m \times n$ | CPU time (second) | Number of nodes | $\mu$ |
|-----------------------|-------------------|-----------------|-----|
|                       | Mean   | Max. | Min. | Mean   | Max. | Min. |   |
| 3                     |        |      |      |        |      |      |   |
| 10 $\times$ 5         | 3.20   | 7.70 | 0.18 | 250    | 600  | 10   | 28 |
| 10 $\times$ 10        | 1.64   | 9.20 | 0.1  | 117.8  | 660  | 3    | 22.3 |
| 25 $\times$ 20        | 13.51  | 28.66| 0.45 | 1257.1 | 2656 | 40   | 26.5 |
| 30 $\times$ 25        | 12.29  | 25.35| 3.83 | 1167.7 | 2242 | 336  | 26 |
| 35 $\times$ 30        | 20.25  | 47.89| 0.44 | 2182.7 | 5892 | 43   | 33.7 |
| 40 $\times$ 35        | 14.29  | 46.48| 0.67 | 1862.8 | 5804 | 71   | 37.6 |
| 45 $\times$ 40        | 22.74  | 54.15| 1.33 | 3193.2 | 7325 | 81   | 29.2 |
| 50 $\times$ 25        | 46.25  | 109.31| 10.57| 6061.5 | 13081| 1405 |   |
| 60 $\times$ 30        | 61.87  | 222.27| 11.70| 8012.6 | 28371| 1675 |   |
| 70 $\times$ 35        | 162.81 | 373.31| 3.35 | 20018.5| 42385| 376  |   |
| 80 $\times$ 40        | 164.48 | 343.94| 4.40 | 21545.1| 42653| 592  |   |
| 90 $\times$ 45        | 208.17 | 462.57| 45.30| 29060.2| 60184| 5880 |   |
| 100 $\times$ 50       | 451.76 | 1127.12| 14.98| 54032.4| 128250| 2106 |   |
| 120 $\times$ 60       | 608.84 | 1015.43| 159.95| 68329.22| 115096| 19820 |   |
| 140 $\times$ 70       | 1477.00| 5250  | 281  | 138    |      |      |   |
| 160 $\times$ 80       | 1477.00| 5250  | 281  | 138    |      |      |   |
| 5                     |        |      |      |        |      |      |   |
| 5 $\times$ 5          | 0.57   | 1.81 | 0.06 | 55.4   | 191  | 3    | 39.3 |
| 10 $\times$ 5         | 2.15   | 6.23 | 0.11 | 262.44 | 725  | 8    | 149 |
| 10 $\times$ 10        | 1.72   | 3.96 | 0.31 | 218.9  | 496  | 38   | 42.5 |
| 20 $\times$ 20        | 4.43   | 12.08| 0.48 | 571.7  | 1463 | 65   | 109.4 |
| 30 $\times$ 30        | 11.01  | 38.81| 2.23 | 1477   | 5250 | 281  | 138 |
| 40 $\times$ 40        | 18.84  | 46.37| 0.22 | 2138   | 4987 | 16   | 125 |
| 50 $\times$ 50        | 41.88  | 103.93| 1.79 | 4677.5 | 10157| 236  | 162.5 |
| 7                     |        |      |      |        |      |      |   |
| 5 $\times$ 5          | 0.26   | 0.56 | 0.06 | 19.5   | 32   | 4    | 36.7 |
| 10 $\times$ 5         | 2.45   | 6.98 | 0.06 | 309.1  | 824  | 3    | 221.7 |
| 10 $\times$ 10        | 1.40   | 2.74 | 0.15 | 199    | 381  | 18   | 110.5 |
| 20 $\times$ 20        | 6.00   | 18.57| 0.52 | 958    | 3104 | 60   | 190.9 |
| 30 $\times$ 30        | 6.71   | 12.86| 1.34 | 953.4  | 1733 | 200  | 174.3 |
| 40 $\times$ 40        | 30.34  | 54.32| 17.82| 4387.5 | 8030 | 2635 | 191.4 |
| 50 $\times$ 50        | 33.4   | 71.02| 2.34 | 4591   | 9401 | 293  | 243.6 |
| 50 $\times$ 25        | 77.31  | 163.62| 14.05| 10553.5| 20943| 2146 |   |
| 60 $\times$ 30        | 55.75  | 164.07| 2.008| 8314.1 | 24890| 351  |   |
| 70 $\times$ 35        | 93.86  | 284.15| 0.404| 12448.8| 32799| 40   |   |
| 80 $\times$ 40        | 176.06 | 350.88| 40.98| 22950.1| 48044| 6468 |   |
| 90 $\times$ 45        | 243.70 | 773.29| 4.33 | 28888.8| 88885| 585  |   |
| 100 $\times$ 50       | 269.34 | 541.84| 51.32| 26847.9| 52410| 5782 |   |
| 110 $\times$ 55       | 517.56 | 847.28| 163.40| 51484.9| 83932| 15147 |   |
| 120 $\times$ 60       | 352.08 | 899.07| 9.58 | 33535.2| 79358| 894  |   |
| 130 $\times$ 65       | 546.83 | 1522.45| 19.69| 50156  | 133845| 1873 |   |
| 140 $\times$ 70       | 799.93 | 2243.55| 45.90| 65041.6| 175900| 5006 |   |

From the computational experiments shown in Table 8, we observe that the proposed method solves small and medium size problems in a reasonable amount of time.
Figure 2: Execution time comparison

In (Figure 6) summarises the average execution time using our method and the Brute Force method, which consists in finding the two efficient sets $X_E$ and $X'_E$ using the algorithm presented in [Chergui & Moulaï, 2008], and then finding the intersection between the two. Notice that in case where $S_E$ is empty the problem has been replaced by another one. We observe that our proposal is faster than Brute Force with 3, 5 and 7 objectives, especially when the number of variables and constraints increases. In fact, when the number of variables and constraints is smaller (5x5, 10x5, 10x10) our method and Brute Force compete with each other, yielding almost the same execution time.

7 Conclusion

In this article, we have presented a novel method to optimize a BIOLFP over the efficient set of a MOILFP problem using a branch-and-cut algorithm. The latter reduces the search domain and thus allows us to avoid exploring all efficient sets $X_E$ and $X'_E$. In addition, the same algorithm can be easily extended in case where there are several utility functions to consider which is a very common real-life case. The experimental results showed that our proposal yields good execution time for small and medium size problems. Furthermore, when compared with Brute Force method, our proposal proved to be better especially when problem size increases.

For future work, we intend to replace linear fractional functions with other non-linear functions.

References

Bajalinov, E. B. (2013). Linear-fractional programming theory, methods, applications and software. Springer Science & Business Media, 84.

Cambini, L. & Martein, A. (1986). A modified version of Martos’ algorithm for the linear fractional problem. Mathematics of Operations Research, 53, 33–44.

Cambini, A., Martein, L. & Stancu-Minasian, I M. (1999). A survey of bicriteria fractional problems. Advanced Modeling and Optimization, 1(1), 9–46.

Charnes, A., & Cooper, W W. (1962). Programming with linear fractional functionals. Naval Research logistics quarterly, 9(3-4), 181–186.

Cherfaoui, Y., & Moulaï, M. (2019). Biobjective optimization over the efficient set of multiobjective integer programming problem. Journal of Industrial and Management Optimization. Advance online publication. doi: dx.doi.org/10.3934/jimo.2019102

Chergui, M. E-A., & Moulaï, M. (2008). An exact method for a discrete multiobjective linear fractional optimization. Journal of Applied Mathematical Decision Science, 2008(760191), 12 pages. http://dx.doi.org/10.1155/2008/760191

Costa, J P. (2007). Computing non-dominated solutions in MOLFP. European Journal of Operational Research, 181(3), 1464–1475.

Drici, W., Ouaïl, F.Z. & Moulaï, M. (2018). Annals of Operations Research, 267(1-2), 135–151.

Ehrgott, M., Hamacher, H.W., Klamroth, K., Nickel, S. Schobel, A. & Wieck, M.M. (1997). A note on the equivalence of balance points and Pareto solutions in multiple-objective programming. Journal of Optimization Theory and Applications, 92(1), 209–212.
Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations research, 9*(6), 849–859.

Jorge, J. M. (2009). An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research, 195*(1), 98–103.

Kornbluth, JSH, & Steuer, R.E. (1981). Multiple objective linear fractional programming. *Management Science, 27*(9), 1024–1039.

Liu, Z. & Ehrgott M. (2018). Primal and dual algorithms for optimization over the efficient set. *Optimization, 67*(10), 1661-1686.

Martos, B. (1964). Hyperbolic Programming. *Naval Research Logistics Quarterly, 11*, 135–155.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science, Kluwer Academic Publishers, Boston, USA 12.

Munson, T. (2007). Mesh shape-quality optimization using the inverse mean-ratio metric. *Mathematical Programming, 110*(3), 561–590.

Philip, J. (1972). Algorithms for the vector maximization problem. *Mathematical programming, 2*(1), 207–229.

Schaible, S. (1981). Fractional programming: applications and algorithms. *European Journal of Operational Research, 7*(2), 111–120.

Stancu-Minasian, I M. (2019). *A ninth bibliography of fractional programming*. Taylor & Francis.

Stancu-Minasian, I M. (2012). *Fractional programming: theory, methods and applications*. Springer Science & Business Media, 409.

Rao, M. R. (1971). Cluster analysis and mathematical programming. *Journal of the American statistical association, 66*(335), 622–626.

Yamamoto, Y. (2002). Optimization over the efficient set: overview. *Journal of Global Optimization, 22*(1-4), 285–317.

Zerdani, O., & Moulaï, M. (2011). Optimization over an integer efficient set of a multiple objective linear fractional problem. *Applied Mathematical Sciences, 5*(50), 2451–2466.