RAIRO-Oper. Res. 58 (2024) 741–758
https://doi.org/10.1051/ro/2024003

RAIRO Operations Research
www.rairo-ro.org

LINEAR FRACTIONAL OPTIMIZATION OVER THE EFFICIENT SET OF MULTI-OBJECTIVE INTEGER QUADRATIC PROBLEM

Ali Bencheikh[1,*] and Mustapha Moulaï[2]

Abstract. In this paper, we introduce an exact algorithm for optimizing a linear fractional utility function over the efficient set of a multi objective integer quadratic problem. The algorithm is based on the "Branch and Cut" principle, which combines the *branching process* to ensure decision variables' integrity and *efficient cuts* built off the non-increasing gradients' directions of objective functions to eliminate inefficient integer solutions. The proposed approach accelerates the convergence to the efficient solution that optimizes the utility function. After presenting and describing the algorithm, a detailed didactic example is illustrated, followed by an experimental study to validate our approach and show computational costs.

Mathematics Subject Classification. 90C29, 90C10, 90C20, 90C32, 90C57.

Received July 18, 2022. Accepted December 30, 2023.

1. Introduction

Multi-objective optimization (MOP) problems encompass multiple conflicting objectives that need to be optimized simultaneously. Unlike single-objective problems, where finding the optimal solution is straightforward, in MOP, it is almost impossible to find a single solution that optimizes all objectives concurrently. Therefore, the goal is to find a set of compromise solutions that represent optimal trade-offs between given objectives. This set of solutions is called *efficient set* or *Pareto set*.

Multi-objective optimization problems have broad applications in fields such as environmental analysis [26], supply chain management [18], design networks [25], portfolio selection [2] and product planning [24]. But in a wide range of MOP applications, restrictions are imposed on the decision variables (*e.g.*, integer variables, binary variables), on the objective functions (e.g non-linear, non-convex) and on the number of objectives to be optimized (bi, tri, or general)-objective. These restrictions highlight the diverse classes of multi-objective optimization.

The fact that finding all efficient solutions is crucial, but can poses two challenges: (1) the abundance of solutions can confuse the decision maker (DM) in selecting the "best" solution from their decision-making viewpoint, (2) as mentioned above, the restrictions and computational viewpoint make generating the complete

Keywords. Multi-objective programming, integer programming, quadratic programming, fractional programming, branch and cut.

[1] USTHB, Dept. of Operations Research, Faculty of Mathematics, BP32, El-Alia 16111, Algeria.

[2] USTHB, LaROMaD Laboratory, Faculty of Mathematics, BP32, El-Alia 16111, Algeria.

*Corresponding author: alibencheikh34@gmail.com

efficient set challenging due to the required numerical efforts. Therefore, to overcome these issues, a utility function that quantifies the decision maker's preferences is associated to be optimized over the efficient set of the MOP in order to target the best solution without generating the whole set of efficient solutions. This approach is known in the literature as the "*optimization over efficient set problems*" and is classified as NP-hard.

A review of the literature indicates significant interest in this problem, despite the difficulty of quantifying the utility function and the non-convexity of the efficient set. For the case of optimizing a linear function over the efficient set of a linear multi-objective problem a number of studies have been conducted, such as: [1, 4, 5, 11, 14, 15].

Nevertheless, in many real-life scenarios, the utility function and/or the multi-objective problem may not always be linear, thus complicating the solution process. Hence, only a few algorithms have been proposed so far. We can quote for example: [20] proposed an algorithm to optimize a nonlinear function on the efficient set of multi-objective integer linear problem (MOILP). Drici *et al.* [10] proposed an exact method to optimize a linear fractional function over the efficient set of a (MOILP). Also, an exact method to optimize an indefinite quadratic function over the efficient set of a (MOILP) was proposed by Moulaï and Drici [17]. And more recently, Chaiblaine and Moulaï [8] where the authors proposed an exact method to optimize a quadratic function over the efficient set of an integer linear fractional multi-objective problem (MOILFP). These recent advances have motivated and inspired us, in this paper, to introduce an exact method to optimize a linear fractional function over the efficient set of a multi-objective integer quadratic problem (henceforth denoted (LF/MOIQP)) that has not been addressed yet.

To show the practicality of our proposed method, we find that in various optimization problems such as in engineering and economics, (DM) consider the minimization of a ratio between physical and/or economical functions, *e.g.*, cost/time, cost/profit, cost/volume, or other quantities that measure the efficiency of a given system [23]. This ratio, which constitutes a fractional function, can considered for minimization over an efficient set of multi-objective problems where these objectives are modeled with convex quadratic functions. Through our review of the literature, we found several real-life optimization problems in which the objective function is modeled by a convex quadratic function: [7] gave modeling to the integer convex quadratic knapsack problem (QKP). Quadri *et al.* [21] considered the problem of integer convex quadratic multidimensional knapsack problems (MDQKP). Djerdjour *et al.* [9] has treated the same problem and added an application in capital budgeting. Gallo *et al.* [12] gave examples in hydro-logical studies and location problems. Shetty and Muthukrishnan [22] have used a quadratic model in parallel projection for the multi-commodity network model. Hua [13] proposed an approximate dynamic programming approach to convex quadratic knapsack problems (adQMP) and mentioned several applications. Bretthauer and Shetty [6] has treated the quadratic resource allocation with generalized upper bounds. Following these examples, we formulate the corresponding multi-objective problem and put forward an application to optimize the DM linear fractional utility function over the efficient set of multi-objective integer quadratic multidimensional knapsack problems.

$$
(LF/MQMKP)\begin{cases}
\min \psi(x) = \dfrac{\sum\limits_{j=1}^{n} p_j x_j + \alpha}{\sum\limits_{j=1}^{n} q_j x_j + \beta} \\
\text{s.t: the efficient set of} \\
\begin{cases}
``Min'' \, f_i(x) = (\sum\limits_{j=1}^{n} \dfrac{1}{2} d_j^i x_j^2 - c_j^i x_j), \quad i = 1,\dots,r \\
\sum\limits_{j=1}^{n} w_{kj} x_j \leq W_k, \quad k = 1,\dots,m. \\
l_j \leq x_j \leq u_j, \quad j = 1,\dots,n. \\
x_j \in \mathbb{Z}, \text{ and } x_j \geq 0, \quad j = 1,\dots,n.
\end{cases}
\end{cases}
$$

where p_j and q_j, and the scalars α and β are coefficients of the (DM) utility function. The decision variable is denoted as x_j. The quadratic cost coefficients are represented by d_j^i, and we assume $d_j^i > 0$ to ensure the convexity of the objective function. The linear cost coefficients are given by c_j. The linear constraints coefficients imposed on the decision variables are specified by w_{kj} and W_k. Without loss of generality, we assume that both $w_{kj} > 0$ and $W_k > 0$.

The principle of proposed method is based on the "Branch and Cut" technique where solutions are found by exploring the branches of the solution tree. The iterative method solves more constrained fractional linear programs in an augmented simplex table, once a solution is obtained, either a branching process is applied to ensure the integrity of the variables, or the optimal solution is integer so it is subjected to an efficiency test to maintain the efficiency. We then use the decreasing directions of the criteria from [19] to build efficient cuts that eliminate non-efficient integer solutions, thereby accelerating convergence towards the optimal efficient solution of the utility function. We also add optimal cuts to eliminate the solutions that do not improve the value of the utility function.

The rest of the paper is organized as follows; Section 2 introduces the problem formulation and notations, followed by the theoretical background and main results in Section 3. However, Section 4 provides a description of the proposed algorithm, illustrated by a detailed example. In Section 5, experimental studies are presented. Finally, the paper concludes with a discussion of perspectives.

2. Problem formulation & notations

2.1. Problem formulation

Let be given r ($r \geq 2$) convex quadratic objective functions of the form:

$$f_i(x) = \frac{1}{2}x^t Q_i x + c_i^t x, \quad i = 1, \dots, r$$

and the set:

$$\mathcal{D} = \mathcal{X} \cap \mathbb{Z}^n \quad \text{where} \quad \mathcal{X} = \{x \in \mathbb{R}_+^n | Ax \leq b\}$$

The multi-objective integer quadratic problem can be defined as follows:

$$(MOIQP) \begin{cases} \text{“}Min\text{”}\ f_i(x), \quad i = 1, \dots, r \\ x \in \mathcal{D} \end{cases}$$

where

- $Q_i \in \mathbb{Z}^{n \times n},\ i = 1, \dots, r$ are positive semi-definite matrices, and $c_i \in \mathbb{Z}^n,\ i = 1, \dots, r$ vectors.
- $A \in \mathbb{Z}^{m \times n}$ matrix, $b \in \mathbb{Z}^m$ vector.
- The set $\mathcal{X}$ is assumed to be closed, bounded and non-empty convex polyhedron.

The problem of optimization over efficient set of (MOIQP) can be stated as follows:

$$(LF/MOIQP) \begin{cases} \min \psi(x) = \dfrac{px^t + \alpha}{qx^t + \beta} \\ x \in \mathcal{X}_{\text{eff}} \end{cases}$$

where:

- $\psi(x)$ is the utility function which is a linear fractional function
- p, q are real n-vectors
- α, β are arbitrary real constants

– $\mathcal{X}_{\text{eff}} \subset \mathcal{D}$ the integer efficient set of the (MOIQP).

Assuming the following conditions hold: $\psi(x)$ is not a strictly positive combination of linear parts of the objective functions. The factor $qx^t + \beta$ is positive for all $x \in \mathcal{X}$, and $\mathcal{X}_{\text{eff}} \neq \mathcal{D}$ ensuring that the problem cannot be solved straightforwardly.

Definition 2.1. A feasible solution x^* is said to be *efficient* in $(MOIQP)$ if, and only if $\nexists\ x \in \mathcal{D}$ such that $f_i(x) \leq f_i(x^*),\ i = 1, \ldots, r$ and $f_i(x) < f_i(x^*)$ for at least one index $i \in \{1, \ldots, r\}$.

The adopted strategy aims to generate the optimal integer solution to the main problem $(LF/MOIQP)$. It is based on resolving a continuous, more constrained linear fractional problem sequence $(LFP)_l$ at each iteration l, where $l \geq 0$. This sequence is defined as follows:

$$(LFP)_l \begin{cases} \min \psi(x) = \dfrac{px^t + \alpha}{qx^t + \beta} \\ x \in \mathcal{X}_l. \end{cases}$$

$\mathcal{X}_0 = \mathcal{X}$ and $\mathcal{X}_l \subset \mathcal{X}$ to be explored at iteration l.

2.2. Notations

To facilitate understanding of the theoretical results, we introduce the following important notations:
We denote the optimal solution of the problem $(LFP)_l$ by $x^{*(l)}$, and we use $\mathcal{B}_l$ and $\mathcal{N}_l$ to denote the index sets of basic and non-basic variables, respectively, of $x^{(l)}$. Simultaneously, x_{opt} stands for the best efficient solution known up to stage l, and ψ_{opt} is the corresponding value of utility function $\psi(x)$. Also, we denote:

– **For utility function**: $\bar{\gamma}_j$ the j^{th} component of growth direction of the vector $\bar{\gamma}$ of the utility function ψ at each iteration with:

$$\bar{\gamma}_j = \psi^2(p_j - \hat{z}_j^1) - \psi^1(q_j - \hat{z}_j^2)$$

where:
  - $\hat{z}_j^1 = p_{\mathcal{B}_l}\mathcal{B}_l^{-1}a_j$; $\hat{z}_j^2 = q_{\mathcal{B}_l}\mathcal{B}_l^{-1}a_j$ and a_j is the j^{th} column of matrix A.
  - $\psi^1 = p^T x^{*(l)} + \alpha$, $\psi^2 = q^T x^{*(l)} + \beta$ and $\psi = \psi^1/\psi^2$.

– **For objective functions**: The functions $f_i,\ i = 1, \ldots, r$ can be written at the neighborhood of $x^{*(l)}$ as:

$$f_i(x) = f_i(x^{*(l)}) + \nabla f_i(x^{*(l)})(x - x^{*(l)}) + \epsilon_i\|x - x^{*(l)}\|$$

where $\nabla f_i(x^{*(l)}) = Q_i x^{*(l)} + c_i$ and $\epsilon_i : \mathbb{R}^n \longmapsto \mathbb{R}$.
The linear expression $\nabla f_i(x^{(l)})(x - x^{(l)})$ determines the non-increasing criterion directions of the edges in the relaxed feasible region, originating from the extreme point $x^{*(l)}$.

– **In the optimal simplex table**: For the basic variables x_k in x, we have the relation $x_k = \hat{b}_{p(k)} - \sum_{j=1}^{n} \hat{a}_{p(k)j}x_j$ where $k \in \mathcal{B}_l$. Here: $p(k)$ indicates the position of x_k in x. Additionally, $\hat{a}_{p(k)j}$ and $\hat{b}_{p(k)}$ represent the updated values in the simplex table of the elements from the constraint matrix A and vector b respectively.

– **Sets and Used cuts**: We define: $\mathcal{N}_l \cap \overline{\{1, n\}}$ as the index set of non basic original variables, $\mathcal{N}_l \backslash \overline{\{1, n\}}$ as the index set of non basic surplus variables, and $\mathcal{B}_l \cap \overline{\{1, n\}}$ as the index set of basic original variables.

The set $\mathcal{H}_l$ identifies non-increasing directions of the criteria $f_i, i = 1, \ldots, r$:

$$\mathcal{H}_l = \left\{ j \in \mathcal{N}_l | \exists i \in \{1, \ldots, r\}; \left(\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right) < 0 \right\}$$
$$\bigcup \left\{ j \in \mathcal{N}_l |; \left(\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right) = 0, \ \forall i \in \{1, \ldots, r\} \right\}$$

where $\nabla f_i(x^{*(l)})_k$ is the k^{th} coordinate of the gradient vector of criterion i in $x^{*(l)}$ and ρ_j is defined as follows:

$$\rho_j = \begin{cases} \nabla f_i(x^{*(l)})_k & \text{if } j \in \mathcal{N}_l \cap \overline{\{1,n\}} \\ 0 & \text{if } j \in \mathcal{N}_l \backslash \overline{\{1,n\}}. \end{cases}$$

Denote by:

$$\bar{f}_i = \rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \text{ for } i = 1, \ldots, r.$$

Our method is based on *Branch and Cut* principle. We use the information to construct the efficient cut to remove all non-efficient integer solutions for $(MOIQP)$. An efficient cut is defined as follows:

$$\sum_{j \in \mathcal{H}_l} x_j \geq 1.$$

Additionally, a second cut is defined to remove solutions that are worse than those already found, given by the inequality:

$$\psi(x) \leq \psi_{\text{opt}}.$$

At the node l, the following subsets are defined:

$$\mathcal{X}^1_{l+1} = \{x \in \mathcal{X}_l \mid \sum_{j \in \mathcal{H}_l} x_j \geq 1\}$$

$$\mathcal{X}^2_{l+1} = \{x \in \mathcal{X}_l \mid \psi(x) \leq \psi_{\text{opt}}\}$$

and

$$\mathcal{X}_{l+1} = \mathcal{X}^1_{l+1} \cup \mathcal{X}^2_{l+1}.$$

3. Theoretical background & main results

In this section, we provide some important definitions and theoretical results that are essential for understanding the subsequent sections.
The following theorem is presented:

Theorem 3.1. *The feasible solution $x^{*(l)}$ is an optimal solution for the problem $(LFP)_l$ if and only if the vector γ is such that $\bar{\gamma}_j \geq 0$ for all $j \in \mathcal{N}_l$. see [16]*

Theorem 3.2. *Supposing that $\mathcal{H}_l \neq \emptyset$ at the current integer solution $x^{*(l)}$. If $x \neq x^{*(l)}$ is an integer efficient solution in domain $\mathcal{X}_l \backslash \{x^{*(l)}\}$, then $x \in \mathcal{X}_{l+1}$.*

Proof. Consider any $x \in \mathcal{X}_l$ such that, $x \neq x^{*(l)}$. Suppose that $x \notin \mathcal{X}_{l+1}$. Then $x \notin \mathcal{X}^1_{l+1}$ and $x \notin \mathcal{X}^2_{l+1}$.

– if $x \notin \mathcal{X}_{l+1}^1$, then $\sum_{j \in \mathcal{H}_l} x_j < 1$ which means that $x_j = 0, \forall j \in \mathcal{H}_l$, x being integer. However, when considering only the n first variables of $(MOIQP)$ of the solution $x^{*(l)}$ and x, we can calculate the following:

$$\nabla f_i(x^{*(l)})(x - x^{*(l)}) = -\nabla f_i(x^{*(l)})x^{*(l)} + \nabla f_i(x^{*(l)})x \quad \forall i \in \{1, \dots, r\}.$$

Furthermore, from the current optimal simplex table at $x^{*(l)}$, we have the aforementioned equation: $x_k = \hat{b}_{p(k)} - \sum\limits_{j \in \mathcal{N}_l} \hat{a}_{p(k)j} x_j$ for all index $k \in \mathcal{B}_l$. Now, we can write the following:

$$\begin{aligned}
\nabla f_i(x^{*(l)})x = & \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k (\hat{b}_{p(k)} - \sum_{j \in \mathcal{N}_l} \hat{a}_{p(k)j} x_j) \\
& + \sum_{j \in \mathcal{N}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_j x_j \qquad \forall i \in \{1, \dots, r\} \\
\nabla f_i(x^{*(l)})x = & \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{b}_{p(k)} \\
& - \sum_{j \in \mathcal{N}_l} \left[\sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} (\nabla f_i(x^{*(l)}))_k \hat{a}_{p(k)j} \right] x_j \\
& + \sum_{j \in \mathcal{N}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_j x_j \qquad \forall i \in \{1, \dots, r\}.
\end{aligned}$$

Besides, we have:

$$\nabla f_i(x^{*(l)})x^{*(l)} = \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{b}_{p(k)}.$$

Thus, we obtain:

$$\begin{aligned}
\nabla f_i(x^{*(l)})(x - x^{*(l)}) = & \sum_{j \in \mathcal{N}_l \cap \overline{\{1,n\}}} \left[\nabla f_i(x^{*(l)})_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \\
& + \sum_{j \in \mathcal{N}_l \setminus \overline{\{1,n\}}} \left[- \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \quad \forall i \in \{1, \dots, r\}.
\end{aligned}$$

Therefore:

$$\nabla f_i(x^{*(l)})(x - x^{*(l)}) = \sum_{j \in \mathcal{N}_l} \left[\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \quad \forall i \in \{1, \dots, r\}.$$

Where:

$$\rho_j = \begin{cases} \nabla f_i(x^{*(l)})_k & \text{if } j \in \mathcal{N}_l \cap \overline{\{1,n\}} \\ 0 & \text{if } j \in \mathcal{N}_l \setminus \overline{\{1,n\}} \end{cases}$$

Using the set $\mathcal{H}_l$ as a subset of the $\mathcal{N}_l$ set, we can write:

$$\begin{aligned}
\nabla f_i(x^{*(l)})(x - x^{*(l)}) = & \sum_{j \in \mathcal{H}_l} \left[\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \\
& + \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} \left[\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \quad \forall i \in \{1, \dots, r\}.
\end{aligned}$$

As it has been assumed that $x_j = 0, \forall j \in \mathcal{H}_l$, then the last expression is reduced to:

$$\nabla f_i(x^{*(l)})(x - x^{*(l)}) = \sum_{j \in \mathcal{N}_l \backslash \mathcal{H}_l} \left[\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \right] x_j \quad \forall i \in \{1, \ldots, r\}.$$

According to the definition of set $\mathcal{H}_l$, for all index $j \in \mathcal{N}_l \backslash \mathcal{H}_l$, we have the following inequality:

$$\rho_j - \sum_{k \in \mathcal{B}_l \cap \overline{\{1,n\}}} \nabla f_i(x^{*(l)})_k \hat{a}_{p(k)j} \geq 0 \quad \forall i \in \{1, \ldots, r\}$$

with at least one strict inequality. Hence:

$$\nabla f_i(x^{*(l)})(x - x^{*(l)}) \geq 0 \quad \forall i \in \{1, \ldots, r\}.$$

Then, we obtain the following inequality:

$$f_i(x^{*(l)}) + \nabla f_i(x^{*(l)})(x - x^{*(l)}) \geq f_i(x^{*(l)}) \quad \forall i \in \{1, \ldots, r\}.$$

From the previous inequality, we conclude that:

$$f_i(x) \geq f_i(x^{*(l)}) \quad \forall i \in \{1, \ldots, r\}.$$

Since there is at least one strict inequality, we deduce that $f(x)$ is dominated by $f(x^{*(l)})$, implying that x is not an efficient solution.

– If $x \notin \mathcal{X}^2_{l+1}$, then $\psi(x) > \psi_{\text{opt}}$. Thus x is not optimal which contradicts the hypothesis.

□

Corollary 3.3. *The constraint* $\sum_{j \in \mathcal{H}_l} x_j \geq 1$ *defines an efficient cut.*

Proof. It is clear that $\sum_{j \in \mathcal{H}_l} x_j \geq 1$ is an efficient valid constraint according to the Theorem 3.2 since all integer efficient solutions in the current domain $\mathcal{X}_l$ satisfy this constraint. However, the current integer solution $x^{*(l)}$ does not, since $x_j = 0$ for all $j \in \mathcal{H}_l$. Thus, the constraint $\sum_{j \in \mathcal{H}_l} x_j \geq 1$ effectively acts as an efficient cut. □

Proposition 3.4. *If* $\mathcal{H}_l = \emptyset$ *at the current integer solution* $x^{*(l)}$*, then* $\mathcal{X}_l \backslash \{x^{*(l)}\}$ *is an explored domain*

Proof. $\mathcal{H}_l = \emptyset$ means that $x^{*(l)}$ is an optimal integer solution for each criterion and, consequently, an ideal point within the domain $\mathcal{X}_l$. Therefore, $\mathcal{X}_l \backslash x^{*(l)}\}$ contains no other efficient solutions. □

Theorem 3.5. *The algorithm converges to an optimal solution of the problem* $(LF/MOIQP)$ *if such a solution exists, in a finite number of iterations*

Proof. Since $\mathcal{D}$ the set of integer solutions of $(MOIQP)$, is a finite and bounded within $\mathcal{X}$, the cardinality of the efficient set $\mathcal{X}_{\text{eff}}$ is also finite. Throughout the algorithm's execution, an optimal integer solution $x^{*(l)}$ is found. If $x^{*(l)}$ is not part of $\mathcal{X}_{\text{eff}}$, an efficient cut is added, or the search for an improved f_{opt} is discontinued. Thus, as stated in the previous theorem and corollary, at least the solution $x^{*(l)}$ is excluded in the analysis of any subsequent sub-problem $(LFP)_k, k > l$, ensuring that no integer efficient solution is overlooked. □

3.1. Efficiency Test

Let $x^{*(l)}$ be an optimal solution of the program $(LFP)_l$, we test the efficiency of the obtained solution by solving the following single objective mathematical program:

$$E(x^{*(l)})\begin{cases}\max\ \varphi=\sum\limits_{i=1}^{r}\varepsilon_i\\ f_i(x)+\varepsilon_i\leq f_i(x^{*(l)}) \qquad i=1,\dots,r\\ \varepsilon_i\geq 0\\ x\in\mathcal{D}\end{cases}$$

the point $x^{*(l)}$ is *efficient* (i.e $x^{*(l)}\in\mathcal{X}_{\text{eff}}$) for the original problem, if the objective function φ is null in $E_\varepsilon(x^{*(l)})$. Else, let $\hat{x}^{(l)}$ be the optimal solution of $E(x^{*(l)})$ then $\hat{x}^{(l)}\in\mathcal{X}_{\text{eff}}$ see [3].

4. Description of method

At each iteration l, we solve the program $(LFP)_l$. The corresponding node l is fathomed when the program $(LFP)_l$ is infeasible or $\mathcal{H}_l=\emptyset$. However, if the optimal solution $x^{*(l)}$ is not integer, we let $x_j=x_j^{*(l)}$ represent one fractional coordinate. Then, the feasible set $\mathcal{X}_l$ is divided into subsets $\mathcal{X}_{l_1}$ and $\mathcal{X}_{l_2}$ by creating two nodes in the search tree with the additional constraints : $x_j\leq\lfloor x_j^{*(l)}\rfloor$ and $x_j\geq\lceil x_j^{*(l)}\rceil$ to construct the program $(LFP)_{l_1}$ and $(LFP)_{l_2}$ respectively, where l_1, and l_2 are greater than l.

In the case, $x^{*(l)}$ is integer, we solve the mathematical program $E(x^{*(l)})$ to test its efficiency. If the solution $x^{*(l)}$ is efficient, we update, if necessary, the value of utility function $\psi(x)$ and the node l is fathomed. Otherwise, let $\hat{x}^{(l)}$ be the efficient solution offered by the test $E(x^{*(l)})$. An update in $\psi(x)$ is made if necessary. After that, we use the non-increasing directions of criteria to define the set $\mathcal{H}_l$ and construct the valid cut $\sum_{j\in\mathcal{H}_l}x_j\geq 1$ to avoid exploring non-efficient regions.

The algorithm steps are detailed in the code below:

4.1. Illustrative example

To illustrate how the algorithm $(LF/MOIQP)$ works, we provide an example to optimize a linear fractional function over the efficient set of a Tri-OIQP problem.

$$(LF/MOIQP)\begin{cases}\text{“Min”}\ \psi(x)=\dfrac{-x_1+5x_2+2x_3+7}{x_1+3x_2+x_3+5}.\\ x\in\mathcal{X}_{\text{eff}}\end{cases}$$

Where: $\mathcal{X}_{\text{eff}}$ is the efficient set of the following (Tri-OIQP):

$$(MOIQP)\begin{cases}\text{“Min”}\ f_i(x)=\frac{1}{2}x^tQ_ix+c_i^tx \quad i=\overline{1,3}\\ Ax\leq b\\ x\in\mathbb{Z}_+^3\end{cases}$$

$$Q_1=\begin{pmatrix}40&43&35\\43&54&41\\35&41&39\end{pmatrix},\ c_1^t=\begin{pmatrix}95&50&-79\end{pmatrix}$$

$$Q_2=\begin{pmatrix}26&32&37\\32&48&52\\37&52&59\end{pmatrix},\ \ c_2^t=\begin{pmatrix}-71&-90&-41\end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 55 & 31 & 41 \\ 31 & 19 & 24 \\ 41 & 24 & 38 \end{pmatrix}, \quad c_3^t = \begin{pmatrix} -40 & -62 & 41 \end{pmatrix}$$

And

$$A = \begin{pmatrix} 9 & 7 & 7 \\ 8 & 3 & 9 \end{pmatrix} \quad x^t = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \quad b = \begin{pmatrix} 26 \\ 21 \end{pmatrix}$$

full example solution steps are detailed in Appendix A, while Figure 1 illustrates and summarizes the resolution process

5. Computational study

In this section, we will conduct a computational study to test and validate the performance of the preceding algorithm. The latter has been implemented in MATLAB 2019b and uses the IBM Cplex 12.8 optimization packages for the quadratic solver. The computations were performed on an Intel (R) Core i7 @ 2.80 GHz computer with 16 GB RAM, running Microsoft Windows 11.

Algorithm 1: A linear fractional optimization over integer efficient set of MOIQP ($LF/MOIQP$).

Ensure: optimal value of the utility function ψ_{opt}, the corresponding efficient solution x_{opt}.
- **Step 1:Initialization**:$\psi_{\text{opt}} = Inf, x_{\text{opt}} = null, l = 0,\ \mathcal{X}_0 = \mathcal{X}$.
- **Step 2:Main Step**:

if $((LFP)_l$==*infeasible*$)$ **then**
  the node l is fathomed.
else
  Let $x^{*(l)}$ be an optimal solution of $(LFP)_l$ **if** $(\psi_{\text{opt}} \leq \psi(x^{*(l)}))$ **then**
    the node l is fathomed
  else
    if $x^{*(l)}$ *is integer* **then**
      go to **Step 2a**
    else
      go to **Step 2b**
    end
    •**Step 2a: Efficiency test** Solve the mathematical program $E(x^{*(l)})$
  end
  if $(x^{(l)})$ *is efficient* **then**
    the node l is fathomed. Update ψ_{opt} if necessary
  else
    Let $\hat{x}^{(l)}$ be an optimal solution of the program $E(x^{*(l)})$. Update ψ_{opt} if necessary, go to **Step 2c**
  end
  • **Step 2b: Branching process** choose one non-integer value from the component of $x^{*(l)}$ say $x_j^{*(l)}$ and divide the feasible set $\mathcal{X}_l$ into two parts $\mathcal{X}_{l_1}$ and $\mathcal{X}_{l_2}$ by adding the constraint $x_j \leq \lfloor x_j^{*(l)} \rfloor$ to obtain the program $(LFP)_{l_1}$, and add the constraint $x_j \geq \lceil x_j^{*(l)} \rceil$ to obtain program $(LFP)_{l_2}$
  • **Step 2c: Efficient cut** determine the set $\mathcal{N}_l$ and the set $\mathcal{H}_l$
  if $(\mathcal{H}_l = \emptyset)$ **then**
    the node l is fathomed
  else
    add the efficient cut to $(LFP)_l$
    if necessary add the cut $\psi(x) \leq \psi_{\text{opt}}$ and, go to **Step 2**
  end
end

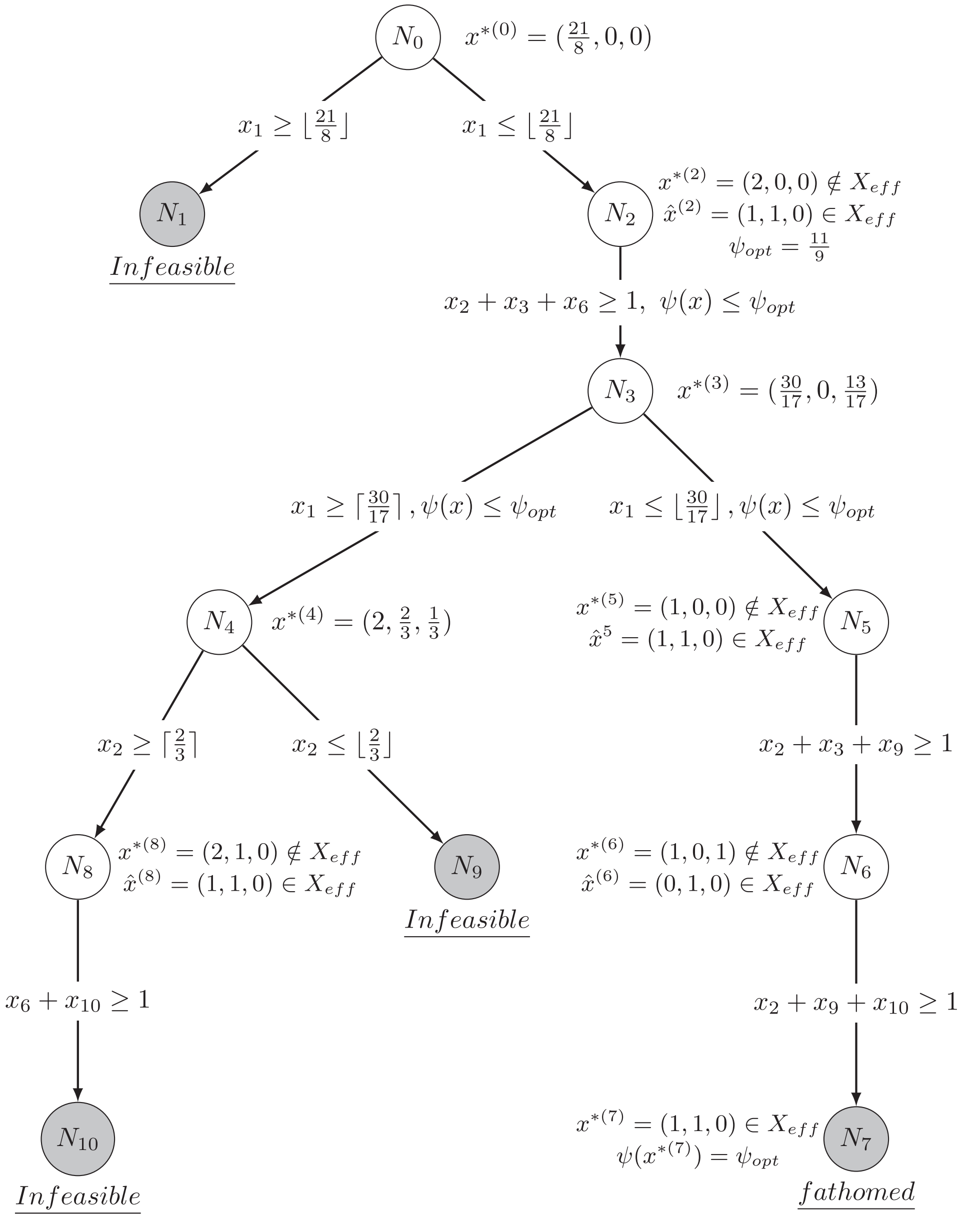

FIGURE 1. Search tree of the example.

The instances were randomly generated with the same distribution for experiments purposes with the following scheme: For objectives, the coefficient of matrices $Q_{i,\, i=1,\dots,r}$ are generated in the interval $\mathcal{U}([1, 5] \times [1, 5])$, which guarantees the generation of semi-definite matrices. For the vectors $c_{i,\, i=1,\dots,r}$ the coefficients are in $\mathcal{U}([-1000, 1000])$. The coefficient of constraints $a_{jk,\, j=1\dots,n,\, k=1\dots,m}$ and the right-hand side of constraints $b_{k,\, k=1\dots,m}$ are in $\mathcal{U}([1, 30])$ and $\mathcal{U}([50, 100])$. For the utility function $\psi(x)$, the vector p and the scalars α we use the same distribution with the linear parts of the quadratic function, however the vector q and the scalar

TABLE 1. Computational results for the algorithms ($LF/MOILP$) and ($LF/MOIQP$).

Number of objectives	Instances size	CPU time(s) of $LF/MOILP$ method		CPU time(s) of $LF/MOIQP$ method	
r	n × m	Average	[min, max]	Average	[min, max]
r = 3	5 × 5	0.296	[0.021, 0.805]	0.281	[0.021, 0.822]
	10 × 10	0.613	[0.031, 2.152]	0.511	[0.047, 1.610]
	15 × 10	4.772	[0.119, 23.67]	5.384	[0.136, 25.69]
	20 × 10	14.27	[0.081, 47.17]	27.64	[0.113, 100.71]
	30 × 15	61.03	[0.223, 163.57]	68.43	[0.621, 159.61]
	30 × 20	27.33	[1.237, 101.02]	29.02	[0.229, 154.57]
	40 × 20	86.92	[22.77, 189.19]	60.79	[4.788, 137.14]
	40 × 25	84.91	[4.162, 189.61]	155.69	[0.534, 599.73]
	50 × 25	230.27	[39.21, 495.81]	260.58	[8.269, 485.34]
	50 × 30	337.43	[39.24, 1940.3]	202.17	[31.39, 395.57]
r = 5	5 × 5	0.051	[0.026, 0.119]	0.078	[0.026, 0.189]
	10 × 10	0.312	[0.031, 1.025]	0.642	[0.030, 2.622]
	15 × 10	1.486	[0.046, 6.204]	0.580	[0.084, 3.202]
	20 × 10	1.937	[0.051, 6.630]	3.606	[0.073, 10.65]
	30 × 15	11.92	[0.095, 29.81]	34.56	[0.300, 101.74]
	30 × 20	9.712	[0.076, 22.23]	18.77	[0.182, 87.93]
	40 × 20	47.23	[0.102, 245.9]	47.32	[0.717, 133.68]
	40 × 25	20.28	[0.821, 96.39]	38.65	[3.931, 228.49]
	50 × 25	61.86	[1.617, 112.67]	61.52	[2.708, 275.53]
	50 × 30	59.09	[6.670, 171.52]	87.27	[0.447, 261.31]
r = 7	5 × 5	0.163	[0.028, 1.335]	0.035	[0.029, 0.045]
	10 × 10	0.088	[0.037, 0.238]	0.152	[0.051, 0.485]
	15 × 10	0.199	[0.032, 0.913]	0.426	[0.038, 2.491]
	20 × 10	0.514	[0.052, 3.255]	1.797	[0.053, 11.32]
	30 × 15	1.183	[0.092, 4.570]	2.159	[0.269, 5.322]
	30 × 20	0.853	[0.061, 6.272]	1.562	[0.149, 4.741]
	40 × 20	4.368	[0.088, 12.71]	45.77	[0.118, 180.14]
	40 × 25	5.322	[0.101, 24.82]	22.88	[0.377, 119.51]
	50 × 25	17.29	[0.351, 62.50]	75.29	[0.767, 301.57]
	50 × 30	11.18	[0.492, 27.65]	30.38	[0.681, 120.24]

β such that $(q^T x + \beta) > 0$ for all $x \in \mathcal{X}$ are generated in $\mathcal{U}([0, 1000])$ and $\mathcal{U}([1, 1000])$. For each (n, m, r) we solve 10 problems.

In the Table 1, we present the obtained numerical results from comparing our method (LF/MOIQP), and that proposed by Drici *et al.* in [10] for optimizing linear fractional over (MOILP) efficient set denoted (LF/MOILP). The same instances were generated for both methods, except that, we consider the matrices $Q_{i,\, i=1,\dots,r}$ null for objectives for the second method. The performance of the proposed algorithms was tested with $r \in \{3, 5, 7\}$ and different sizes problems, and we report the average CPU time (s), the minimum and the maximum for each instance.

Based on the results presented in Table 1, we can notice that the proposed method is effective and performs well. For $r = 3$ and $r = 5$, our method yields solution times that are comparable to the second method. Moreover, for $r = 7$ with 50 decision variables and 30 constraints, the proposed method consistently achieves the optimal solution in less than 1 minute on average. On the other hand, it is important to highlight that applying the efficiency test on the obtained integer solutions during the algorithm process necessitates solving a Mixed Integer Quadratically Constrained Problem (MIQCP) repeatedly. The non-linearity of the quadratic constraints intensifies the computational complexity of the problem, contributing to the disparity in solution

TABLE 2. Computational results: explored nodes and efficient cuts used for the algorithm ($LF/MOIQP$).

Number of objectives	Instances size	Number of explored nodes		Number of efficient cuts		μ
r	n × m	Average	[min, max]	Average	[min, max]	average
r = 3	10 × 10	79.7	[3, 639]	16.3	[0, 144]	0.0910
	15 × 10	276.8	[3, 1074]	47.6	[0, 207]	0.0831
	20 × 10	566.8	[3, 2998]	89.8	[0, 485]	0.0755
	30 × 15	560	[4, 1712]	67.4	[1, 197]	0.0511
	30 × 20	969.1	[9, 3319]	115.5	[0, 410]	0.0581
	40 × 20	977.9	[1, 2970]	104.1	[0, 301]	0.0609
	40 × 25	2724.4	[3, 17202]	358	[2, 2753]	0.0490
	50 × 25	1351.9	[13, 7342]	98.3	[4, 473]	0.0410
	50 × 30	661.4	[5, 2100]	52.2	[0, 177]	0.0385
r = 5	10 × 10	21	[2, 137]	3.2	[0, 24]	0.0171
	15 × 10	49.1	[3, 268]	5.3	[0, 39]	0.0184
	20 × 10	58.5	[1, 400]	5.7	[0, 51]	0.0115
	30 × 15	504.	[2, 1736]	57.6	[1, 187]	0.0149
	30 × 20	74.8	[3, 233]	9.2	[0, 38]	0.0171
	40 × 20	541.9	[5, 1146]	54.3	[0, 109]	0.0226
	40 × 25	396.3	[3, 1376]	34.5	[0, 151]	0.0203
	50 × 25	1184.4	[4, 4409]	105.6	[1, 434]	0.0157
	50 × 30	534.3	[28, 2085]	29.7	[1, 116]	0.0177
r = 7	10 × 10	13.1	[2, 81]	1.1	[0, 6]	0.0099
	15 × 10	21.2	[2, 97]	1.2	[0, 8]	0.0069
	20 × 10	107.3	[3, 743]	12.3	[0, 84]	0.0074
	30 × 15	77.7	[3, 360]	8.7	[0, 37]	0.0041
	30 × 20	194.1	[1, 1657]	18.5	[0, 170]	0.0043
	40 × 20	298.8	[6, 862]	19.8	[0, 78]	0.0096
	40 × 25	670.7	[4, 2224]	47.1	[1, 179]	0.0104
	50 × 25	459.5	[5, 2171]	34.9	[0, 140]	0.0084
	50 × 30	558.1	[13, 4778]	34.7	[0, 287]	0.0049

time observed for $r = 7$. However, one advantageous aspect is that the optimization package utilized in this study has been specifically developed to efficiently handle MIQCPs. As a result, it significantly reduces the computational burden associated with the problems addressed in this paper.

Furthermore, it is worth mentioning that both methods require less time to solve a larger number of objectives. This efficiency can be attributed to the fact that when dealing with a greater number of objectives, the set of efficient solutions expands. Consequently, the method requires fewer efficient cuts to reach the optimal solution. In contrast, when the number of objectives is smaller, the method relies on a greater number of efficient cuts to eliminate inefficient solutions. This necessitates the utilization of more additional variables and the creation of more nodes, resulting in significantly increased time consumption. This observation can be seen in the results presented in the Table 2.

In the Table 2, we present the average, minimum, and maximum number of explored nodes and number of efficient cuts. Additionally, we introduce the parameter $\mu = \tilde{\mathcal{X}}_{\text{eff}}/\mathcal{X}_{\text{eff}}$, which represents the ratio of visited efficient solutions ($\tilde{\mathcal{X}}_{\text{eff}}$) over the total number of efficient solutions ($\mathcal{X}_{\text{eff}}$) of the (MOIQP). This ratio provides insights into the algorithm's convergence towards the optimal solution.

From the data in Table 2, it is apparent that fewer efficient cuts are utilized for a larger number of objectives. The efficacy of these cuts is reflected in their ability to eliminate inefficient solutions. For instance, when

TABLE 3. Computational results for the problem $(LF/MQMKP)$.

Number of objectives	Dimensions	CPU time(s)		Number of possible combinations	
r	n × m	Average	[min, max]	Average	[min, max]
r = 2	5 × 2	34.38	[5.41, 81.21]	395.8	[49, 841]
	5 × 5	12.94	[1.63, 36.81]	120.8	[24, 291]
	10 × 5	824.98	[35.29, 2113.57]	5521.2	[255, 14099]
	15 × 10	837.72	[256.48, 2324.63]	3960.6	[1195, 10863]
	20 × 10	2780.30	[359.41, 9986.01]	9871.2	[1148, 35721]
	20 × 15	8029.69	[1825.13, 20898.89]	21397	[6635, 50547]
r = 3	5 × 2	20.49	[0.27, 58.91]	105.4	[2, 236]
	5 × 5	64.07	[8.97, 125.75]	387	[38, 751]
	10 × 5	725.33	[68.01, 1427.82]	2135	[207, 4031]
	15 × 10	1046.16	[15.16, 2768.83]	3790	[60, 10498]
	20 × 10	2847.85	[1073.43, 4840.91]	8054.8	[3647, 14983]
	20 × 15	687.94	[324.87, 1447.86]	1789.6	[906, 3409]
r = 5	5 × 2	40.34	[6.381, 84.11]	493.2	[32, 1379]
	5 × 5	8.884	[1.526, 22.45]	78.2	[17, 199]
	10 × 5	4017.45	[78.44, 14125.91]	21391.4	[490, 77472]
	15 × 10	725.34	[101.28, 1246.56]	2146	[233, 4288]
	20 × 10	2586.81	[297.54, 5096.56]	8779.2	[1237, 19950]
	20 × 15	6940.33	[198.78, 28399.45]	18165.4	[503, 75055]
r = 7	5 × 2	28.213	[7.003, 97.42]	525.8	[47, 1937]
	5 × 5	7.711	[3.669, 18.97]	117.8	[43, 340]
	10 × 5	511.07	[13.33, 1845.23]	3299.2	[103, 11669]
	15 × 10	1011.54	[59.16, 2308.81]	5016	[295, 12386]
	20 × 10	834.40	[164.43, 1342.87]	3056.8	[704, 5629]
	20 × 15	1690.41	[245.89, 3409.47]	4858.4	[671, 9898]

considering $r = 3$, the ratio μ does not exceed 6.08% from the total number of listed efficient solutions of (MOIQP). Similarly, for $r = 5$ and $r = 7$ it does not go beyond 1.72% and 0.73% respectively.

In Table 3, we present numerical results for the problem $(LF/MQMKP)$, based on data from [9] and [21]. The instances tested were uniformly random and uncorrelated.

For the $(MQMKP)$ objective functions, parameters are set as: $d_j \in \mathcal{U}([5,10])$, $c_j \in \mathcal{U}([10,20])$, $w_{jk} \in \mathcal{U}([10,40])$, $W_k \in \mathcal{U}([100,400])$ with $j = 1,\dots,n$ and $k = 1,\dots,m$. Decision variable bounds are $l_j = 0$ and $u_j = 10$ for $j = 1,\dots,n$. For the utility function, p and α are drawn from $\mathcal{U}([-20,20])$, while q and β are from $\mathcal{U}([0,20])$ and $\mathcal{U}([1,20])$ respectively. The table also lists the number of objectives $r \in \{2,3,5,7\}$, knapsack dimensions, CPU time in seconds, and combinations of integer nodes in the solution tree.

Based on the results presented in Table 3, it is evident that the algorithm requires significantly more time to solve than what is observed in Table 1. This outcome is expected, given that the problem (LF/MQMKP), is NP-hard. In certain instances, the solution time exceeds 5 hours, with over 75,000 potential knapsack combinations explored. This implies that the efficiency test needs to be applied at least 75,000 times, thereby posing a considerable challenge for the algorithm's computational capabilities.

Nevertheless, it is noteworthy that, on average, the algorithm manages to provide an optimal solution within a reasonable time-frame, especially considering the complexity of the problem, particularly for $r = 7$.

6. Conclusion

Through this paper, we have introduced an exact algorithm designed to optimize a fractional linear utility function over the efficient set of multi-objective integer quadratic problem. It is worth noting that this algorithm is versatile and can be easily adapted to optimize a linear utility function with just a few essential modifications. The algorithm can be easily modified to optimize a linear utility function by making a few necessary changes. Looking forward, our goal is to enhance the algorithm's efficiency in terms of computational time and its robustness, particularly in the context of real-life applications. Moreover, we are committed to extending the scope of our algorithm to address more complex decision problems, such as optimizing a quadratic utility function, thereby contributing further to the field.

Appendix A. Full example resolution steps

Initialization: we solve the program $(LFP)_0$, the first optimal solution obtained $x^{*(0)} = (\frac{21}{8}, 0, 0)$. The results are summarized in the following Table:

Table A.1. Optimal simplex table for node N_0.

$\mathcal{B}_1$	x_2	x_3	x_5	RHS
x_1	3/8	9/8	1/8	21/8
x_4	29/8	−25/8	−9/8	19/8
$\bar{\gamma}_j$	59/2	195/8	3/2	

Table A.2. Optimal simplex table for node N_2.

$\mathcal{B}_2$	x_2	x_3	x_6	RHS
x_1	0	0	1	2
x_4	7	7	−9	8
x_5	3	9	−8	5
$\bar{\gamma}_j$	20	9	12	
$\bar{f}_1$	136	−9	−175	
$\bar{f}_2$	−26	33	19	
$\bar{f}_3$	0	123	−70	

As the solution $x^{*(0)}$ is not an integer, we apply the branching process and, two nodes N_1 and N_2 are created with the following constraints:

$$N_1 : x_1 \geq \lceil \frac{21}{8} \rceil$$

$$N_2 : x_1 \leq \lfloor \frac{21}{8} \rfloor$$

For N_1: the added constraint $x_1 \geq \lceil \frac{21}{8} \rceil$ renders the augmented program $(LFP)_1$ unfeasible; therefore, the node is fathomed.
For N_2: the constraint $x_1 \leq \lfloor \frac{21}{8} \rfloor$ is added to the Table A.1. Solving $(LFP)_2$ yields the Table A.2, which presents the solution $x^{*(2)} = (2, 0, 0)$ with $\psi(x^{*(2)}) = \frac{5}{7}$.

TABLE A.3. Optimal simplex table for node N_3.

$\mathcal{B}_3$	x_2	x_5	x_7	RHS
x_1	$-6/17$	$1/17$	$9/17$	$30/17$
x_3	$11/17$	$1/17$	$-8/17$	$13/17$
x_4	$96/17$	$-16/17$	$-25/17$	$81/17$
x_6	$6/17$	$-1/17$	$-9/17$	$4/17$
x_8	$7/153$	$13/153$	$236/153$	$373/153$
$\bar{\gamma}_j$	$118/17$	$6/17$	$195/17$	

TABLE A.4. Optimal simplex table for node N_4.

$\mathcal{B}_4$	x_5	x_7	x_9	RHS
x_1	0	0	-1	2
x_2	$-1/6$	$-3/2$	$-17/6$	$2/3$
x_3	$1/6$	$1/2$	$11/6$	$1/3$
x_4	0	7	16	1
x_6	0	0	1	0
x_8	$5/54$	$26/18$	$7/54$	$65/27$
$\bar{\gamma}_j$	$5/3$	$74/3$	$59/3$	

The solution $x^{*(2)}$ is integer, we test the efficiency by solving $E(x^{*(2)})$. The solution is not efficient; however the efficient solution generated by the test is $\hat{x}^{(2)} = (1, 1, 0)$ with $\psi(\hat{x}^{(2)}) = \frac{11}{9}$. We update $\psi_{\text{opt}} = \frac{11}{9}$, $x_{\text{opt}} = (1, 1, 0)$, the set $\mathcal{H}_2 = \{2, 3, 6\}$, the efficient cut $x_2 + x_3 + x_6 \geq 1$ is applied with the cut $\psi(x) \leq \psi_{\text{opt}}$. The solution $x^{*(3)} = (\frac{30}{17}, 0, \frac{13}{17})$. The results of resolution of $(LFP)_3$ are shown in Table A.3.

The solution $x^{*(3)}$ is not integer, the branching process is also applied with the following constraints:

$$N_4 : x_1 \geq \lceil \frac{30}{17} \rceil, \psi(x) \leq \psi_{\text{opt}}$$

$$N_5 : x_1 \leq \lfloor \frac{30}{17} \rfloor, \psi(x) \leq \psi_{\text{opt}}.$$

After solving the corresponding problem $(LFP)_4$ and $(LFP)_5$, we obtain the following solution: $x^{*(4)} = (2, \frac{2}{3}, \frac{1}{3})$ with $\psi(x^{*(4)}) = \frac{27}{25}$, and $x^{*(5)} = (1, 0, 0)$ with $\psi(x^{*(5)}) = 1$, respectively. The result are shown in the Tables A.4 and A.5.

As $\psi(x^{*(5)}) < \psi(x^{*(4)})$, we first explore the node N_5, the solution $x^{*(5)}$ is integer, we test its efficiency by solving $E(x^{*(5)})$. The test yields the solution $\hat{x}^{(5)} = (1, 1, 0)$ with $\psi(\hat{x}^{(5)}) = \frac{11}{9}$. the set $\mathcal{H}_5 = \{2, 3, 9\}$ is identified and, we apply the efficient cut $x_2 + x_3 + x_9 \geq 1$

$$N_6 : x_2 + x_3 + x_9 \geq 1.$$

The results of $(LFP)_6$ are presented in Table A.6:

The obtained solution $x^{*(6)} = (1, 0, 1)$ with $\psi(x^{*(6)}) = \frac{8}{7}$. Since the solution $x^{*(6)}$ is integer, it's subjected to the efficiency test $E(x^{*(6)})$; the test offer the efficient solution $\hat{x}^{(6)} = (0, 1, 0)$ with $\psi(\hat{x}^{(6)}) = \frac{3}{2}$. We define the set $\mathcal{H}_6 = \{2, 9, 10\}$ and add the efficient cut $x_2 + x_9 + x_{10} \geq 1$.

$$N_7 : x_2 + x_9 + x_{10} \geq 1.$$

The results of $(LFP)_7$ are summarized in Table A.7:

TABLE A.5. Optimal simplex table for node N_5.

$\mathcal{B}_5$	x_2	x_3	x_9	RHS
x_1	0	0	1	1
x_4	7	7	-9	17
x_5	3	9	-8	13
x_6	0	0	-1	1
x_7	-1	-1	-1	0
x_7	4/3	7/9	20/9	4/3
$\bar{\gamma}_j$	12	6	12	
$\bar{f}_1$	93	-44	-135	
$\bar{f}_2$	-58	-4	45	
$\bar{f}_3$	-31	82	-15	

TABLE A.6. Optimal simplex table for node N_6.

$\mathcal{B}_6$	x_2	x_9	x_{10}	RHS
x_1	0	1	0	1
x_3	1	1	-1	1
x_4	0	-16	7	10
x_5	-6	-17	9	4
x_6	0	-1	0	1
x_7	0	0	-1	1
x_8	5/9	13/9	7/9	5/9
$\bar{\gamma}_j$	5	9	6	
$\bar{f}_1$	139	-165	-5	
$\bar{f}_2$	-61	-47	55	
$\bar{f}_3$	-127	-176	120	

TABLE A.7. Optimal simplex table for node N_7.

$\mathcal{B}_7$	x_9	x_{10}	x_{11}	b
x_1	1	0	0	1
x_2	0	0	-1	1
x_3	0	-2	1	0
x_4	-16	7	0	10
x_5	-11	15	-6	10
x_6	-1	0	0	1
x_7	0	-1	0	1
x_8	8/9	2/9	5/9	0
$\bar{\gamma}_j$	8	2	5	
$\bar{f}_1$	-325	-153	150	
$\bar{f}_2$	23	106	-58	
$\bar{f}_3$	-34	224	-118	

TABLE A.8. Optimal simplex table for node N_8.

$\mathcal{B}_8$	x_6	x_7	x_{10}	b
x_1	1	0	0	2
x_2	0	0	−1	1
x_3	1	−1	1	0
x_4	−16	7	0	1
x_5	−17	9	−6	2
x_8	13/9	7/9	5/9	20/9
x_9	1	0	0	0
$\bar{\gamma}_j$	10	10	10	
$\bar{f}_1$	−250	32	158	
$\bar{f}_2$	−98	85	−63	
$\bar{f}_3$	−248	147	−128	

The solution $x^{*(7)} = (1, 1, 0)$ is optimal integer solution, and $\psi(x^{*(7)}) = \psi_{\text{opt}}$. Hence, the node N_7 is fathomed. Back to exploring the node N_4: the solution $x^{*(4)} = (2, \frac{2}{3}, \frac{1}{3})$ is not integer, the branching process generates the constraints:

$$N_8 : x_2 \geq \lceil \frac{2}{3} \rceil$$

$$N_9 : x_2 \leq \lfloor \frac{2}{3} \rfloor.$$

In node N_9, the added constraints make the augmented Table A.4 infeasible, consequently the node N_9 is fathomed. However, for the node N_8 after adding the constraint to the Table A.4, we obtain the solution $x^{*(8)} = (2, 1, 0)$, the results of $(LFP)_8$ are presented in the following Table A.8:

The solution $x^{*(8)}$ is an optimal integer solution, thus it is submitted to efficiency test by solving $E(x^{*(8)})$. We obtain the efficient solution $\hat{x}^8 = (1, 1, 0)$ with $\psi_{\text{opt}} = f(\hat{x}^8) = \frac{11}{9}$.
The set $\mathcal{H}_8 = \{6, 10\}$, and we apply the efficient cut $x_6 + x_{10} \geq 1$

$$N_{10} : x_6 + x_{10} \geq 1.$$

In the node N_{10} the added constraints make the program $(LFP)_{10}$ infeasible, so the node N_{10} is fathomed. All tree nodes are fathomed, hence no further improvement could be attempted. However $x_{\text{opt}} = (1, 1, 0)$ and $\psi_{\text{opt}} = \frac{11}{9}$.

Acknowledgements. The authors are grateful to Doctor Ilias Badaoui, Doctor Hadjer Moulaï and anonymous referees for their substantive comments and suggestions that improved the quality and the presentation of the paper. Authors work was supported by the Direction Générale de la Recherche Scientifique et du Dveloppement Technologique (DGRSDT) Grant ID: C0656104.

References

[1] M. Abbas and D. Chaabane, Optimizing a linear function over an integer efficient set. *Eur. J. Oper. Res.* **174** (2006) 1140–1161.

[2] F. Abdelaziz, B. Aouni and R. El Fayedh, Multi-objective stochastic programming for portfolio selection. *Eur. J. Oper. Res.* **177** (2007) 1811–1823.

[3] H. Benson, Existence of efficient solutions for vector maximization problems. *J. Optim. Theory Appl.* **26** (1978) 569–580.

[4] H. Benson, Optimization over the efficient set. *J. Math. Anal. Appl.* **98** (1984) 562–580.

[5] N. Boland, H. Charkhgard and M. Savelsbergh, A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *Eur. J. Oper. Res.* **260** (2017) 904–919.

[6] K. Bretthauer and B. Shetty, Quadratic resource allocation with generalized upper bounds. *Oper. Res. Lett.* **20** (1997) 51–57.

[7] K. Bretthauer and B. Shetty, The nonlinear knapsack problem–algorithms and applications. *Eur. J. Oper. Res.* **138** (2002) 459–472.

[8] Y. Chaiblaine and M. Moulaï, An exact method for optimizing a quadratic function over the efficient set of multiobjective integer linear fractional program. *Optim. Lett.* (2021) 1–15.

[9] M. Djerdjour, K. Mathur and H. Salkin, A surrogate relaxation based algorithm for a general quadratic multi-dimensional knapsack problem. *Oper. Res. Lett.* **7** (1988) 253–258.

[10] W. Drici, F. Ouail and M. Moulaï, Optimizing a linear fractional function over the integer efficient set. *Ann. Oper. Res.* **267** (2018) 135–151.

[11] J. Ecker and J. Song, Optimizing a linear function over an efficient set. *J. Optim. Theory Appl.* **83** (1994) 541–563.

[12] G. Gallo, P. Hammer and B. Simeone, Quadratic knapsack problems. *Comb. Optim.* (1980) 132–149.

[13] Z. Hua, B. Zhang and L. Liang, An approximate dynamic programming approach to convex quadratic knapsack problems. *Comput. Oper. Res.* **33** (2006) 660–673.

[14] J. Jorge, An algorithm for optimizing a linear function over an integer efficient set. *Eur. J. Oper. Res.* **195** (2009) 98–103.

[15] B. Lokman, Optimizing a linear function over the nondominated set of multiobjective integer programs. *Int. Trans. Oper. Res.* (2019).

[16] B. Martos, Nonlinear Programming: Theory and Methods (1975).

[17] M. Moulaï and W. Drici, An indefinite quadratic optimization over an integer efficient set. *Optimization* **67** (2018) 1143–1156.

[18] B. Nepal, L. Monplaisir and O. Famuyiwa, A multi-objective supply chain configuration model for new products. *Int. J. Prod. Res.* **49** (2011) 7107–7134.

[19] F. Ouaïl and M. Chergui, A branch-and-cut technique to solve multiobjective integer quadratic programming problems. *Ann. Oper. Res.* **267** (2018) 431–446.

[20] M. Ozlen, M. Azizoğlu and B. Burton, Optimising a nonlinear utility function in multi-objective integer programming. *J. Glob. Optim.* **56** (2013) 93–102.

[21] D. Quadri, E. Soutif and P. Tolla, Exact solution method to solve large scale integer quadratic multidimensional knapsack problems. *J. Comb. Optim.* **17** (2009) 157–167.

[22] B. Shetty and R. Muthukrishnan, A parallel projection for the multicommodity network model. *J. Oper. Res. Soc.* **41** (1990) 837–842.

[23] I. Stancu-Minasian, Fractional Programming: Theory, Methods and Applications. Springer Science & Business Media (2012).

[24] R. Wang and T. Liang, Application of fuzzy multi-objective linear programming to aggregate production planning. *Comput. Ind. Eng.* **46** (2004) 17–41.

[25] B. Zahiri, R. Tavakkoli-Moghaddam, M. Mohammadi and P. Jula, Multi-objective design of an organ transplant network under uncertainty. *Transp. Res. Part E: Logist. Transp. Rev.* **72** (2014) 101–124.

[26] A. Zamboni, F. Bezzo and N. Shah, Spatially explicit static model for the strategic design of future bioethanol production systems. 2. Multi-objective environmental optimization. *Energy Fuels* **23** (2009) 5134–5143.