Branch and cut method for solving integer indefinitequadratic bilevel programs with multiple objectives atthe upper level

Fatima Fali, Mustapha Moulai

https://doi.org/10.1590/SciELOPreprints.11577

Submitted on: 2025-03-26
Posted on: 2025-03-31 (version 1)
(YYYY-MM-DD)

Noname manuscript No.
(will be inserted by the editor)

Branch and cut method for solving integer indefinite quadratic bilevel programs with multiple objectives at the upper level

Fatima Fali · Mustapha Moulaï

Received: date / Accepted: date

Abstract This paper proposes an exact method to solve an integer indefinite quadratic bilevel problem with multiple objectives at the upper level, where the objective functions at both levels are a product of two linear functions. The suggested algorithm uses a branch and cut algorithm based on a multiobjective integer linear problem obtained by replacing the indefinite quadratic objectives of the upper level by their two linear functions and the classical branch and bound technique for integer decision variables. Then, the integer solutions obtained are tested for optimality of the lower level problem by using a library IBM CPLEX 12.8 for C++ programs.
The integer indefinite quadratic bilevel programming problem with single objective at both levels is solved in the first step, based on the dantzig cut. The second phase explorates with the efficient cut to provide the set of efficient solutions without listing the whole integer domain.
After the presentation of the algorithm, a numerical example and computational experiments are provided.

Keywords Multi-objective programming · bilevel programming · integer programming · quadratic programming · linear programming · branch and cut

Mathematics Subject Classification (2020) 90C29 · 90C10 · 90C20 · 90C57

1 Introduction

Two types of decision makers with hierarchical structure exist in a bilevel programming problem: a leader and a follower, called upper and lower level decision

Fatima Fali
USTHB University, LaROMaD laboratory, Faculty of Mathematics,
BP 32 Bab-Ezzouar, El-Alia 16111, Algiers, Algeria
E-mail: ffali@usthb.dz; faliffatima@gmail.com https://orcid.org/0000-0002-7233-0936

Mustapha Moulaï
USTHB University, LaROMaD laboratory, Faculty of Mathematics,
BP 32 Bab-Ezzouar, El-Alia 16111, Algiers, Algeria
E-mail: mmoulai@usthb.dz https://orcid.org/0000-0002-2013-356X

makers respectively. In this problem, the feasible region of the upper level problem is implicitly determined by the optimization problem of the lower level. These areas have been widely studied in the literature [11, 26, 12, 9] and often appear in many practical problems [4].

Moreover, in [25, 28] tackled the integer indefinite quadratic bilevel programming problem: Narang and Arora [25] proposed an algorithm for solving such a problem with bounded variables, recently Maachou and Moulaï[28] presented an algorithm where the optimal solution of the integer indefinite quadratic problem belongs to the efficient solutions set of the corresponding bicriteria problem.

Furthermore, in real-world, decision-making processes can have multiple conflicting objectives to optimize, for example, in transportation system planning and traffic management, a situation like this can be modelled like a bilevel programming problem with multiple objectives at the upper level [27]. In multiobjective optimization problems, a feasible solution is said to be efficient if it is impossible to increase the value of any objective function without decreasing the values of the other objective function in a multiobjective optimization problem, in contrast to a single optimisation problem [8].

In multiple objective bilevel problems, some researchers have presented algorithms for the linear bilevel multiobjective programming problems [2, 3, 6, 14–16, 18, 20]. However, non linear bilevel multiobjective programming problems have not been treated much in the literature: Shi and Xia [22, 23] presented an interactive method, Abo-Sinna [1] and Osman *et al.* [19] proposed the usage of fuzzy set theory for convex problems, Teng *et al.*[24] and Deb and Sinha [10] provided some evolutionary algorithms, Bonnel and Morgan suggested a method based on a penalty approach in [5] and Zheng and Wan [30] proposed an exact penalty method. Concerning the indefinite quadratic bilevel programming problem with multiple objectives, Arora [29] developed two algorithms to find an efficient solution, one of them dedicated to the problem with multiple objectives at the upper level and the other dedicated with multiple objectives at both levels.

An algorithm has been proposed by Fali and Moulaï in [13] that generates the whole integer efficient solutions of a linear fractional integer bilevel maximization problem with multiple objectives at the upper level based on the branch and cut algorithm.

This study draws inspiration from the method described in [13], an exact algorithm for solving an indefinite quadratic integer bilevel maximization problem with multiple objectives at the upper level, is proposed. The algorithm developed strengthen the branch and bound procedure by adding a dantzig cut [28] for finding the optimal bilevel solution of the integer indefinite quadratic bilevel programming problem with single objective at both levelsand Chergui and Moulaï's efficient cut [8] to the branching process for finding the efficient set.

In the following section, some basic definitions and notations are defined. The third section is devoted how to solve multiple objectives indefinite quadratic programs. The fourth section is dedicated to the presentation of the exact algorithm for solving integer indefinite quadratic bilevel programs with multiple objective functions at the upper level, while an illustrative example and numerical experiments are reported in the fifth and sixth sections respectively. The last section of the paper is dedicated to the conclusion.

2 Definitions and notations

The integer indefinite quadratic bilevel programs with multiple objective functions at the upper level can be formulated as:

$$(IQMOBP)\begin{cases}\max\limits_{(x,y)}\ F^1(x,y)=(c_1^1x+c_2^1y+\alpha^1)(d_1^1x+d_2^1y+\beta^1)\\ \max\limits_{(x,y)}\ F^2(x,y)=(c_1^2x+c_2^2y+\alpha^2)(d_1^2x+d_2^2y+\beta^2)\\ \dots\\ \max\limits_{(x,y)}\ F^k(x,y)=(c_1^kx+c_2^ky+\alpha^k)(d_1^kx+d_2^ky+\beta^k)\\ s.t\\ A_1^1x+A_2^1y\le b^1\\ \text{where y solves}\\ \begin{cases}\max\limits_{y}\ g(x,y)=(p^1x+p^2y+\omega)(q^1x+q^2y+\delta)\\ \text{s.t}\\ A_1^2x+A_2^2y\le b^2\\ x\ge 0, y\ge 0, \text{ and integer}\end{cases}\end{cases} \tag{1}$$

where

$$x\in\mathbb{R}^{n_1}\quad and\quad y\in\mathbb{R}^{n_2} \tag{2}$$

are variables controlled by the upper level and the lower level decision maker, respectively;

$$F,g:\mathbb{R}^n\longmapsto\mathbb{R},\ n=n_1+n_2 \tag{3}$$

are the upper level and lower level objective functions, respectively;
$k\ge 2$ is the number of objective functions of the leader.
$A_1^1\in\mathbb{R}^{m_1\times n_1}$; $A_2^1\in\mathbb{R}^{m_1\times n_2}$; $b^1\in\mathbb{R}^{m_1}$
$A_1^2\in\mathbb{R}^{m_2\times n_1}$; $A_2^2\in\mathbb{R}^{m_2\times n_2}$;$b^2\in\mathbb{R}^{m_2}$
$c_1^i, d_1^i, p^1, q^1\in\mathbb{R}^{n_1}$; $c_2^i, d_2^i, p^2, q^2\in\mathbb{R}^{n_2}$; the constants $\omega,\delta,\alpha^i,\beta^i\in\mathbb{R}$, $i\in\{1,\dots,k\}$

2.1 Constraint region

The constraint region of the $(IQMOBP)$, noted S, is defined by all constraint of the leader and follower as follows:

$$S=\{(x,y): A_1^1x+A_2^1y\le b^1, A_1^2x+A_2^2y\le b^2, x\ge 0, y\ge 0\} \tag{4}$$

Throughout this article, it is assumed that S is not empty and $(c_1^ix+c_2^iy+\alpha^i)$ and $(d_1^ix+d_2^iy+\beta^i)$ are positive over S for all $i\in\{1,\dots,k\}$ as well as $(p^1x+p^2y+\omega)$ and $(q^1x+q^2y+\delta)$.
Let $\mathbb{Z}^n$ be the set of integer numbers and $\mathcal{D}$ be the set of integer solutions over S. So, $\mathcal{D}=S\cap\mathbb{Z}^n$.
In a bilevel problem, the upper level selects its choice first, then the follower reacts by selecting an y that optimizes its objective function.

2.2 **Lower level problem**

For a given x, the lower level problem $(LIQP)$ can be written as:

$$(LIQP(x)) \begin{cases} \max\limits_{y} \; g(x,y) = (p^1x + p^2y + \omega)(q^1x + q^2y + \delta) \\ A_2^2 y \leq b^2 - A_1^2 x \\ y \geq 0, \text{ and integer} \end{cases} \tag{5}$$

$M(x)$ is the set of optimal solutions to the lower level problem:

$$M(x) = arg \max_{y} \; \{g(x,y) : A_2^2 y \leq b^2 - A_1^2 x; \text{ for a given } x\} \tag{6}$$

2.3 **Inducible region**

The feasible region of the upper level decision maker, called inducible region IR, is implicitly defined by the lower level optimization problem:

$$\text{IR} = \{(x, y^*) \in \mathcal{D}, y^* \in M(x)\}$$

Therefore, the follower has impact on the realisable region of the upper level and on the outcoms for his single or multiple objective functions. Particulary, if $M(x)$ is not singleton and as the leader may not force the follower to choose a response, the leader cannot reach its maximum over IR.

To avoid this situation, it is assumed that the optimal solution of the lower level is unique. In other words, for each value x of the upper level problem, there will be a unique solution y to the lower level problem [7].
All points of IR are bilevel feasible solutions. Taking into account the previous notations, the bilevel problem formulated in (1) can be written as follows:

$$(IQMOBP) \begin{cases} \max\limits_{z} \; F^1(z) = (c^1z + \alpha^1)(d^1z + \beta^1) \\ \max\limits_{z} \; F^2(z) = (c^2z + \alpha^2)(d^2z + \beta^2) \\ \dots \\ \max\limits_{z} \; F^k(z) = (c^kz + \alpha^k)(d^kz + \beta^k) \\ s.t \quad z \in \text{IR} \end{cases} \tag{7}$$

where $z = (x, y) \in \mathbb{R}^{n_1+n_2}$, $c^i = (c_1^i, c_2^i) \in \mathbb{R}^{n_1+n_2}$ and $d^i = (d_1^i, d_2^i) \in \mathbb{R}^{n_1+n_2}$. Many approaches for solving the bilevel multiobjective optimization problem ([2], [6], [16], [20]) use the concept of efficiency. The efficient solutions are bilevel feasible solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest of objective functions.

The objective is to seek to obtain the best of the bilevel feasible solutions depending to all objective functions, called efficient solutions to the $IQMOBP$, according to the following definition:

Definition 1 [6] A bilevel feasible solution $z \in$ IR is an efficient solution of $(IQMOBP)$ problem if and only if there exists no $\overline{z} \in$ IR, such that $F^i(\overline{z}) \geq F^i(z)$, for all $i \in \{1, \dots, k\}$ and $F^i(\overline{z}) > F^i(z)$ for at least one $i \in \{1, \dots, k\}$. Otherwise, z is not efficient and its criterion vector $F(z)$ is dominated by the criterion vector $F(\overline{z})$, where $F(z) = F^i(z)_{i=1,\dots,k}$.

2.4 Technique to solve indefinite quadratic problem

Consider the following indefinite quadratic problem:

$$(IQP)\begin{cases}\max\limits_{z} \; F^1(z)=f^1(z)f^2(z)\\ s.t \quad z\in\mathcal{D}\end{cases} \tag{8}$$

It is known Nash (1950) [17] that an indefinite quadratic problem (IQP) is equivalent to the following problem:

$$(IQP)\begin{cases}\max\limits_{z} \; F^1(z)=f^1(z)f^2(z)\\ s.t \quad z\in\mathcal{X}_E\subset\mathcal{D}\end{cases} \tag{9}$$

where $\mathcal{X}_E$ is the set of efficient solutions of bicriteria linear program (10) obtained by replacing the indefinite quadratic objective by its two linear factors as follows:

$$(BLP)\begin{cases}\max\limits_{z} \; f^1(z)\\ \max\limits_{z} \; f^2(z)\\ s.t \quad z\in\mathcal{D}\end{cases} \tag{10}$$

Definition 2 [21] A feasible solution $z\in D$ is an efficient solution of (10) problem, if there is no other $\overline{z}\in D$ such that $f^1(z)\le f^1(\overline{z})$ and $f^2(z)<f^2(\overline{z})$ or $f^1(z)<f^1(\overline{z})$ and $f^2(z)\le f^2(\overline{z})$.

Proposition 1 *[21] An optimal solution of problem (8), denoted by z^*, is an efficient solution of problem* (10) *and therefore its corresponding image in the criterion space, denoted by $F^1(z^*)$ where $F^1(z^*)$, is a non dominated point.*

To generate the optimal solution of the problem (8), means to scan all efficient solutions of problem (10), noted $\mathcal{X}_E$, see [8]. The method is based on solving, at each step l, a linear problem (ULP_l) $(l\in\mathbb{N})$, defined by

$$(ULP_l)\begin{cases}\max\limits_{z} \; f^1(z)=(c^1z+\alpha^1)\\ s.t \quad z\in S_l\end{cases} \tag{11}$$

where $S_0=S$ and S_l is a subset of the original set S.
Let z^{*l} be the first integer solution obtained through solving problem (ULP_l) by using, eventually, the branching process well known in branch and bound technique. The sets I_l and N_l denote, respectively, the index set of basic variables and the index set of non basic variables of $z^{*(l)}$. Let $\overline{c}_j^i$ be the j^{th} component of the reduced gradient vector $\overline{c}^i$ for each objective function f^i, $i\in\{1,2\}$ at the last simplex table. That methodology is a branch and cut method so that, at each step l of the algorithm, a cut (13) is added to the initial domain to avoid non efficient solutions. To do so, the following index sets at the integer solution $z^{*(l)}$ are defined:

$$H_l=\{j\in N_l|\overline{c}_j^2>0\}\cup\{j\in N_l|\overline{c}_j^i=0,\forall i\in\{1,2\}\} \tag{12}$$

and the inequality

$$\sum_{j\in H_l} z_j \geq 1 \tag{13}$$

Therefore, the set S_{l+1} is obtained by applying (23) to $z^{*(l)}$

$$S_{l+1} = \left\{ z \in S_l : \sum_{j\in H_l} z_j \geq 1 \right\} \tag{14}$$

The algorithm [8] generating an integer optimal solution of problem (8) is presented in the following steps. The nodes in the tree structure are treated according to the backtracking principle.

Algorithm 1: Algorithm for solving problem (8)

Result: $z^{*(l)}$ is the integer optimal solution of problem (8)
while *there is a non-fathomed node* l **do**
  solve the problem (11);
  if (16) *has an optimal solution* $z^{*(l)}$ **then**
    if $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ *is integer* **then**
      if *the vector* $F(z^{*(l)})$ *is not dominated by the vector* $F(z)$ *for any* $z \in \mathcal{X}_E$ **then**
        $\mathcal{X}_E = \mathcal{X}_E \cup (z^{*(l)})$
      if *there is a solution* $z \in \mathcal{X}_E$ *such that* $F(z^{*(l)})$ *dominates* $F(z)$ **then**
        $\mathcal{X}_E = \mathcal{X}_E \setminus \{z\} \cup \{(z^{*(l)})\}$
      Construct the set H_l;
      if $H_l = \emptyset$ **then**
        Fathom the node l
      else
        Add the cut (13) to the successor of l;
    else
      Choose an index j such as $z_j^{*(l)}$ is fractional. Then, split the program (16) into two sub programs, by adding respectively the constraints $z_j \leq \left\lfloor z_j^{*(l)} \right\rfloor$ and $z_j \geq \left\lfloor z_j^{*(l)} \right\rfloor + 1$ to obtain (ULP_{l_1}) and (ULP_{l_2}) ($l_1 \geq l+1$, $l_2 > l+1$ and $l_1 \neq l_2$);
  else
    Fathom the node l;

Theorem 1 *[8] The proposed algorithm generates the integer optimal solution of the program (8) in a finite number of iterations, if such solutions exist.*

2.5 **Technique to solve indefinite quadratic bilevel problem with a single objective at each level**

The integer indefinite quadratic bilevel problem with a single objective function at each level can be formulated as:

$$(BP)\begin{cases}\max\limits_{(x,y)} \; F^1(x,y) = (c_1^1 x + c_2^1 y + \alpha^1)(d_1^1 x + d_2^1 y + \beta^1) \\ s.t \quad (x,y) \in \mathrm{IR}\end{cases} \tag{15}$$

Definition 3 [25] A bilevel feasible solution $(x, y) \in \mathrm{IR}$ is an optimal solution of problem (15) if and only if there exists no $(\overline{x}, \overline{y}) \in \mathrm{IR}$, such that $F^1(\overline{x}, \overline{y}) > F^1(x, y)$.

To generate the optimal solution of the problem (15), the proposed method is based on solving, at each step l, an indefinite quadratic programming problem $(UQP_l)(l \in \mathbb{N})$ at the upper level,using the above algorithme 1, defined by:

$$(UQP_l)\begin{cases}\max \; F^1(z) = (c^1 z + \alpha^1)(d^1 z + \beta^1) \\ s.t \quad z \in S_l\end{cases} \tag{16}$$

$S_0 = S$ and S_l is a subset of the original set S.
Solve program (UQP_0) first. This phase consists in solving problem (16) at step l and the integer optimal solution $z^{*(l)} = (x^{*(l)}, y^{*(l)})$, it is tested for optimality of the lower level problem (5). To do so, let x be fixed at $x = x^{*(l)}$ and let $\hat{y}$ be an integer optimal solution of the lower level problem (5), such a problem can be transformed into a second-order cone program and can be solved by a library IBM CPLEX 12.8 for C++ programs. If $y^{*(l)} = \hat{y}$, $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ is an integer bilevel optimal solution of problem (15).
If this solution is not a bilevel feasible solution of problem (15), a cut (17) is added and repeat the process.
To do this, the cut defined from the simplex table corresponding to the integer optimal solution $z^{*(l)}$ of the problem (16) is:

$$\sum_{j \in N_l} z_j \geq 1 \tag{17}$$

The dantzig cut (17) is added to the current simplex table, which allows to eliminate the integer optimal solution $z^{*(l)}$ of problem (16) and find the second best integer solution of the problem (16)[13].
Therefore, the set S_{l+1}^1 is obtained by applying (17) to $z^{*(l)}$

$$S_{l+1}^1 = \left\{ z \in S_l : \sum_{j \in N_l} z_j \geq 1 \right\} \tag{18}$$

It follows that the cut (17) eliminates from S_l only the integer optimal solution of problem (16).

This phase finishes when the problem is infeasible or the integer optimal solution of problem (15) is found.

The algorithm which determines an integer optimal solution of problem (15) is presented in the following steps:

Algorithm 2: Algorithm for solving problem (15)

Result: Let $z^{*(l)}$ be the integer optimal solution of problem (15)
Initialization $l = 0$, $S_0 = S$, $S_E = \emptyset$;
while $S_E = \emptyset$ **do**
  Solve the problem (16) and find its integer optimal solution;
  if *(16) has an integer optimal solution* $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ **then**
    Put $x = x^{*(l)}$ and solve the problem (5).
    Let $\hat{y}$ be the integer optimal solution of problem (5);
    if $y^{*(l)} = \hat{y}$ **then**
      $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ is the integer optimal solution of problem (15), $S_E = \{z^{*(l)}\}$
    else
      Add the cut (17) to the successor of l;
  else
    The integer optimal solution of problem (15) does not exist;

Theorem 2 *The proposed algorithm generates integer an optimal integer solution of the program (15) in a finite number of iterations, if such a solution exist.*

Proof As S is assumed to be bounded, the set $\mathcal{D}$ is finite and bounded. Since the feasible region is bounded and is truncated at each step by branching process and repeated applications of cut (17) corresponding to the respective integer optimal solution of problem $(UQP_k), k > l$, the whole of S is scanned in such a way that points once scanned and truncated cannot reappear, leading to convergence of the procedure in a finite number of steps. In the case where, the decision set S_l does not contain any optimal integer solution, the subtree rooted at node l is explored and all nodes are fathomed after finitely many steps.

3 Technique to solve multiple objectives indefinite quadratic programs

Consider the multiobjective indefinite quadratic problem defined by:

$$(IQMOP) \begin{cases} \max_z \; F^1(z) = (c^1 z + \alpha^1)(d^1 z + \beta^1) \\ \max_z \; F^2(z) = (c^2 z + \alpha^2)(d^2 z + \beta^2) \\ \dots \\ \max_z \; F^k(z) = (c^k z + \alpha^k)(d^k z + \beta^k) \\ s.t \quad z \in \mathcal{X}_u \end{cases} \tag{19}$$

where $\mathcal{X}_u$ represents the integer efficient set of the following (20) problem defined by:

$$(LMOP) \begin{cases} \max\limits_{(z)} \ f^1(z) = (c^1 z + \alpha^1) \\ \max\limits_{(z)} \ f^2(z) = (d^1 z + \beta^1) \\ \max\limits_{(z)} \ f^3(z) = (c^2 z + \alpha^2) \\ \max\limits_{(z)} \ f^4(z) = (d^2 z + \beta^2) \\ ... \\ \max\limits_{(z)} \ f^{2k-1}(z) = (c^k z + \alpha^k) \\ \max\limits_{(z)} \ f^{2k}(z) = (d^k z + \beta^k) \\ s.t \quad z \in \mathcal{D} \end{cases} \tag{20}$$

Proposition 2 *An efficient solution of problem (19), denoted by z^*, is an efficient solution of problem* (20) *and its corresponding image in the criterion space, denoted by $F(z^*)$ where $F(z^*)$, is a non dominated by the criterion vector $F(\overline{z})$, where $F(z) = F^i(z)_{i=1,...,k}$.*

Proof Suppose that z^* is an efficient solution of Problem (19) but it is not an efficient solution of Problem (20). By definition, this implies that there must exist a feasible solution denoted by $z \in$ IR that dominates z^*. In other words, we must have $f^i(z^*) \leq f^i(z)$, for all $i \in \{1, ..., 2k\}$ and $f^i(z^*) < f^i(z)$ for at least one $i \in \{1, ..., 2k\}$. Therefore, we must have that $0 < F^i(z^*) \leq F^i(z)$, for all $i \in \{1, ..., k\}$ and $F^i(z^*) < F^i(z)$ for at least one $i \in \{1, ..., k\}$. Consequently, z^* cannot be an efficient solution of Problem (19) (a contradiction).

4 Technique to solve integer indefinite quadratic bilevel programs with multiple objectives at the upper level

Let S_E be the set of efficient solutions of the problem (1) initialized to the empty set.
The first phase of the algorithm starts by finding the optimal solution of the problem (15), using the above algorithm (2).
Once an integer optimal solution of problem (15) is achieved, the second phase of the algorithm generate the set of efficient solutions without enumerating the whole integer domain the strategy consists in solving program (21), defined by

$$(ULP_l) \begin{cases} \max\limits_{z} \ f^1(z) = (c^1 z + \alpha^1) \\ s.t \quad z \in S_l^2 \end{cases} \tag{21}$$

at each step of the algorithm. Then, to update the reduced gradient vector $\overline{c}^i$, $i \in \{1, ..., 2k\}, 2k$ lines are added to basic simplex table and reduced costs are calculated with respect to the corresponding basis.
In case this optimal solution is not integer, let $z_j^{*(l)}$ be one component of $z^{*(l)}$ so that $z_j^{*(l)} = \nu_j$, where ν_j is a fractional number. The node l of the tree is then separated in two nodes which are imposed by the additional constraints $z_j^{*(l)} \leq \lfloor \nu_j \rfloor$

and $z_j^{*(l)} \geq \lfloor \nu_j \rfloor + 1$, where $\lfloor \nu_j \rfloor$ indicates the greatest integer less than ν_j.
As soon as an integer solution $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ is found at the new node l, it is tested for optimality of the lower level problem (5). To do so, let x be fixed at $x = x^{*(l)}$ and let $\hat{y}$ be an integer optimal solution of the lower level problem (5), such a problem can be transformed into a second-order cone program and can be solved by a library IBM CPLEX 12.8 for C++ programs. If $y^{*(l)} = \hat{y}$, $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ is an integer bilevel feasible solution of problem (1).
When an integer bilevel feasible solution is obtained, it is compared to the set of efficient solutions S_E and the set S_E is updated. The cut (23) is established and added to the current simplex table, which allows avoiding the non-efficient solutions of problem (1) and determining a new integer solution.
To do so, the set of indexes of all increasing directions of the criteria for each node l, is defined as follows:

$$H_l = \{j \in N_l | \exists i \in \{2, ..., k\}, \overline{c}_j^i > 0\} \cup \{j \in N_l | \overline{c}_j^i = 0, \forall i \in \{1, ..., k\}\} \tag{22}$$

and the inequality

$$\sum_{j \in H_l} z_j \geq 1 \tag{23}$$

is called efficient cut, which removes only non efficient integer solutions [8].
Therefore, the set S_{l+1}^2 is obtained by applying (23) to $z^{*(l)}$

$$S_{l+1}^2 = \left\{ z \in S_l : \sum_{j \in H_l} z_j \geq 1 \right\} \tag{24}$$

where S_{l+1}^2 is the complement of S_{l+1}^1 to S_{l+1}:

$$S_{l+1} = S_{l+1}^1 \cup S_{l+1}^2$$

However, if no improvement of the criteria can be done along the remaining domain ($H_l = \emptyset$), the node l is fathomed.

This phase finishes when all the created nodes are fathomed. A node l of the tree is fathomed if the corresponding program is infeasible or the set H_l is empty.

The algorithm generating the set of efficient solutions of problem (1) is presented in the following steps. The nodes in the tree structure are treated according to the backtracking principle.

Algorithm 3: Algorithm for solving problem (1)

Result: S_E is the efficient set of problem (1)
Initialization By using the above algorithm 2, we generate the integer optimal solution $z^{*(l)}$ of problem (15), start with $S_E = \{z^{*(l)}\}$;
while *there is a non-fathomed node* l **do**
  solve the problem (21);
  if (21) *has an optimal solution* $z^{*(l)}$ **then**
    if $z^{*(l)} = (x^{*(l)}, y^{*(l)})$ *is integer* **then**
      Put $x = x^{*(l)}$ and solve the problem (5).
      Let $\hat{y}$ be the integer optimal solution of problem (5);
      if $y^{*(l)} = \hat{y}$ **then**
        if *the vector* $F(z^{*(l)})$ *is not dominated by the vector* $F(z)$ *for any* $z \in S_E$ **then**
          $S_E = S_E \cup (z^{*(l)})$
        if *there is a solution* $z \in S_E$ *such that* $F(z^{*(l)})$ *dominates* $F(z)$ **then**
          $S_E = S_E \setminus \{z\} \cup \{(z^{*(l)})\}$
      Construct the set H_l;
      if $H_l = \emptyset$ **then**
        Fathom the node l
      else
        Add the cut (23) to the successor of l;
    else
      Choose an index j such as $z_j^{*(l)}$ is fractional. Then, split the program (21) into two sub programs, by adding respectively the constraints $z_j \leq \left\lfloor z_j^{*(l)} \right\rfloor$ and $z_j \geq \left\lfloor z_j^{*(l)} \right\rfloor + 1$ to obtain $(UIQP_{l_1})$ and $(UIQP_{l_2})$ ($l_1 \geq l+1$, $l_2 > l+1$ and $l_1 \neq l_2$);
  else
    Fathom the node l;

The following theoretic [8,13] tools show that the algorithm generates efficient integer solutions of program (1) in a finite number of iterations.

Theorem 3 *[8,13] Assume that $H_l \neq \emptyset$ at the current integer solution $z^{*(l)}$. If z is an integer efficient solution in domain $S_l \setminus \{z^{*(l)}\}$, then $z \in S_{l+1}$.*

Corollary 1 *[8,13] Assume that $H_l \neq \emptyset$ at the current integer solution $z^{*(l)}$. Then constraint $\sum_{j \in H_l} z_j \geq 1$ defines an efficient cut.*

Proposition 3 *[8,13] If $H_l = \emptyset$ at the current integer solution $z^{*(l)}$, then $S_l \setminus \{z^{*(l)}\}$ is an explored domain.*

Theorem 4 *The proposed algorithm generates efficient integer solutions of the program $(IQMOBP)$ in a finite number of iterations, if such a solution exist.*

Proof As S is assumed to be bounded, the set $\mathcal{D}$ is finite and bounded, consequently the cardinal efficient set S_E is a finite number. Each time an optimal integer solution $z^{*(l)}$ is calculated the efficient cut is added if $z^{*(l)} \notin S_E$ or not. Thus, according to the above theorem and corollary, at least the solution $z^{*(l)}$ is eliminated when one studies any sub-problem $(ULIQP_k), k > l$. In the case that there is no optimal integer solution in the decision set S_l, the subtree rooted at node l is explored and all nodes are fathomed after finitely many steps.

5 Illustrative example

To illustrate the use of the algorithm, consider the following problem:

$$(IQMOBP)\begin{cases} \max\limits_{x_1} \ F^1(x_1,y_1,y_2) = (x_1+2y_2+3)(3y_1+2) \\ \max\limits_{x_1} \ F^2(x_1,y_1,y_2) = (2x_1+y_1+2)(y_2+1) \\ s.t \\ 3x_1+y_1+2y_2 \quad \leq 5 \\ y_1+y_2 \qquad \leq 3 \\ \text{where } (y_1,y_2) \text{ solves} \\ \qquad \begin{cases} \max g(x_1,y_1,y_2) = (y_2+1)(x_1+y_1+y_2+3) \\ s.t \\ x_1+2y_1+y_2 \quad \leq 2 \\ 3y_1+2y_2 \qquad \leq 6 \\ x_1 \geq 0, y_1 \geq 0, y_2 \geq 0, integers \end{cases} \end{cases} \tag{25}$$

Initialization: Put $(y_1, y_2) = (x_2, x_3)$ and $S = \{(x_1, x_2, x_3) : 3x_1 + x_2 + 2x_3 \leq 5, x_2 + x_3 \leq 3, x_1 + 2x_2 + x_3 \leq 2, 3x_2 + 2x_3 \leq 6, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0\}$.
Start with $S_E = \emptyset$, $l = 0$.

$$(IQP_0)\begin{cases} \max \ F^1(x_1,x_2,x_3) = (x_1+2x_3+3)(3x_2+2) \\ s.t \\ 3x_1+x_2+2x_3 \quad \leq 5 \\ x_2+x_3 \qquad \leq 3 \\ x_1+2x_2+x_3 \quad \leq 2 \\ 3x_2+2x_3 \qquad \leq 6 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, integers \end{cases} \tag{26}$$

5.1 **Phase I**

Node 0:

1. The solution mentioned in Table 1 is integer after solving the program IQP_0: $z^{*(0)} = (0, 0, 2)$.
   (a) Put $x_1 = 0$ in the lower level problem, the integer optimal solution found is $(y_1^{*(0)}, y_2^{*(0)}) = (0, 2)$.

   (b) Since $(y_1^{*(0)}, y_2^{*(0)}) = (x_2^{*(0)}, x_3^{*(0)})$, the solution $z^{*(0)} = (0, 0, 2)$ is integer efficient solution of our problem $(IQMOBP)$, then the set $S_E = \{(0, 0, 2)\}$ with $F(z^{*(0)}) = (14, 6)$ and $H_1 = \{1, 2\}$.

5.2 **Phase II**

1. **Node 1:** After adding the efficient cut $x_1 + x_2 \geq 1$ to Table 1. Table 2 is obtained, as well as the optimal solution $z^{*(1)} = (1, 0, 1)$ with $F(z^{*(1)}) = (12, 8)$, the set $S_E = \{(0, 0, 2), (1, 0, 1)\}$ and $H_1 = \{2, 8\}$.

   (a) $z^{*(1)} = (1, 0, 1)$, put $x_1 = 1$ in the lower level problem, the integer optimal solution found is $(0, 1)$. Since $(0, 1) = (0, 1)$, the solution $z^{*(1)} = (1, 0, 1)$ is a bilevel efficient solution of our problem (25).

   (b) **Node 2:** After adding the efficient cut $x_2+x_8 \geq 1$ to Table 2. The solution found, $z^{*(2)} = (\frac{5}{3}, 0, 0)$, is not integer. So, the node is divided in two nodes.

   (a) **Node 4:** After adding the constraint $x_1 \leq \lfloor \frac{5}{3} \rfloor$, the optimal solution is obtained $z^{*(3)} = (1, \frac{1}{2}, 0)$. This solutions too is not integer. So, the node is divided in two nodes.

      i. **Node 5:** After adding the constraint $x_2 \geq \lceil \frac{1}{2} \rceil$, Table 2 is obtained, with optimal solution $z^{*(1)} = (0, 1, 0)$ and $F(z^{*(1)}) = (15, 3)$ and $H_1 = \{2, 8\}$.

         A. $z^{*(1)} = (0, 1, 0)$, put $x_1 = 0$ in the lower level problem, the integer optimal solution found is $(0, 2)$.
         Since $(1, 0) \neq (0, 2)$, the solution $z^{*(1)} = (0, 1, 0)$ is not a bilevel efficient solution of our problem (25).

      ii. **Node 6:** After adding the constraint $x_2 \leq \lfloor \frac{1}{2} \rfloor$. Table 2 is obtained with optimal solution $z^{*(1)} = (0, 0, 1)$ and $F(z^{*(1)}) = (8, 4)$ when is dominated and $H_1 = \emptyset$, then the node is fathomed.

      iii. $z^{*(5)} = (0, 0, 1)$, put $x_1 = 0$ in the lower level problem, the integer optimal solution found is $(0, 2)$. Since $(0, 1) \neq (0, 2)$, the solution $z^{*(1)} = (0, 0, 1)$ is not a bilevel efficient solution of our problem (25).

   (b) **Node 3:** After adding the constraint $x_1 \geq \lceil \frac{5}{3} \rceil$ the problem becomes infeasible, then the node is fathomed.

2. **Node 8:** After adding the efficient cut $x_2 + x_9 \geq 1$ to Table 4, the problem becomes infeasible, then the node is fathomed.

Then the set of all integer efficient bilevel solutions for the main problem (25) is $\{(1, 0, 1), (0, 0, 2)\}$

Table 1 Simplex table for node 0.

B_0	x_1	x_2	x_6	Rhs
x_4	1	-3	-2	1
x_5	-1	-1	-1	1
x_3	1	2	1	2
x_7	-2	-1	-2	2
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-1	-4	-2	-7
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	0	3	0	-2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	2	1	0	-2
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	-1	-2	-1	-3

Table 2 Simplex table for node 1.

B_1	x_2	x_6	x_8	Rhs
x_4	-4	-2	1	0
x_5	0	-1	-1	2
x_3	1	1	1	1
x_7	1	-2	-2	4
x_1	1	0	-1	1
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-3	-2	-1	-6
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	3	0	0	-2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-1	0	2	-4
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	-1	-1	-1	-2

Table 3 Simplex table for node 5.

B_2	x_9	x_{10}	x_{11}	Rhs
x_4	-2	-3	-1	2
x_5	-1	0	-1	3
x_3	1	0	0	0
x_7	-2	0	-3	6
x_1	0	1	0	1
x_8	-1	0	-1	1
x_2	0	0	1	0
x_6	-1	-1	-2	1
$c^1_{1j} + c^1_{2j} - z^{1(1)}_j$	-2	-1	0	-4
$d^1_{1j} + d^1_{2j} - z^{1(2)}_j$	0	0	-3	-2
$c^2_{1j} + c^2_{2j} - z^{2(1)}_j$	0	-2	-1	-4
$d^2_{1j} + d^2_{2j} - z^{2(2)}_j$	-1	0	0	-1

Table 4 Simplex table for node 6.

B_2	x_6	x_9	x_{11}	Rhs
x_2	0	0	-1	1
x_5	-1	-1	1	2
x_3	1	1	0	0
x_7	-2	-2	3	3
x_1	0	-1	2	0
x_8	0	-1	1	0
x_4	-2	1	-5	4
x_{10}	0	1	-2	1
$c^1_{1j} + c^1_{2j} - z^{1(1)}_j$	-2	-1	-2	-3
$d^1_{1j} + d^1_{2j} - z^{1(2)}_j$	0	0	3	-5
$c^2_{1j} + c^2_{2j} - z^{2(1)}_j$	0	2	-3	-3
$d^2_{1j} + d^2_{2j} - z^{2(2)}_j$	-1	-1	0	-1

To summarize the proposed Branch & Cut method throughout this example, a tree representing states of the nodes during the process is presented (Figure (1)).

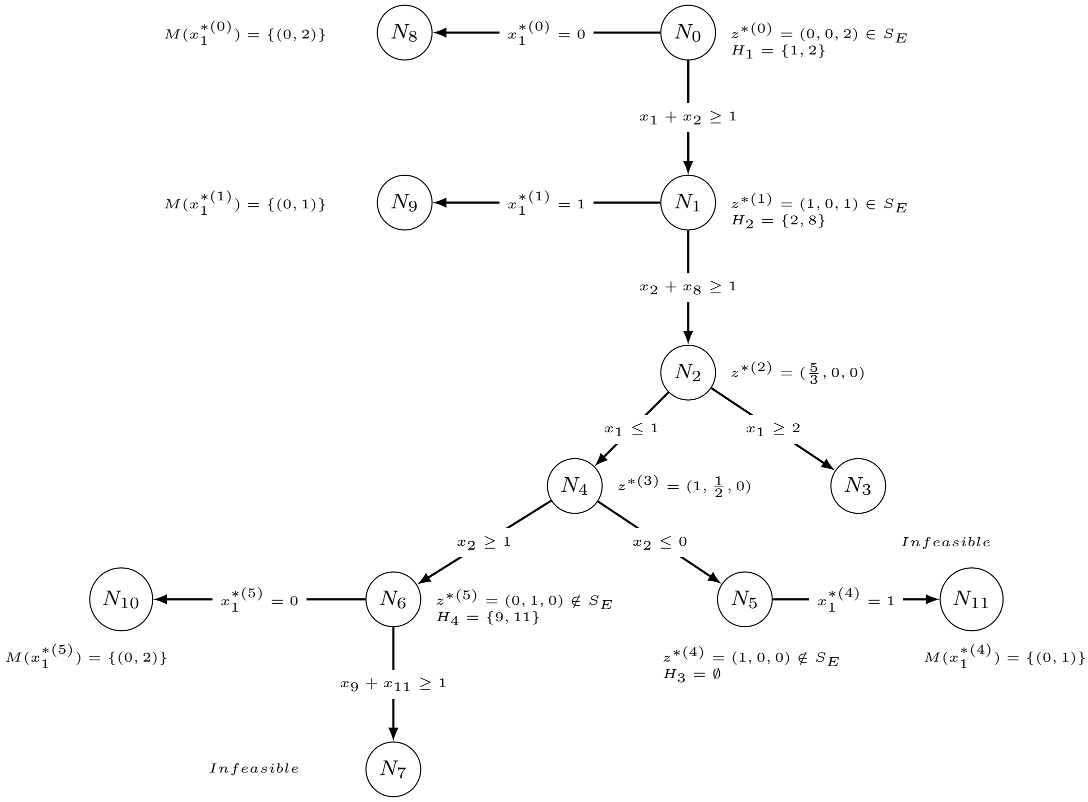

Fig. 1 Tree representing states of nodes during Branch & Cut algorithm.

6 Numerical experiments

The method was implemented with the compiler Visual C++ 14.0 and the program was run on an Intel(R) Core (TM) i5 CPU 2.4 GHZ with 4 GB of the main memory and the operating system Windows10. the method was tested with randomly generated instances. The data is randomly generated by an integer uniform distribution in the interval [1, 10] for constraints coefficients and [50,100] for right hand side coefficient bj. The vectors $c_1^i, d_1^i, c_2^i, d_2^i, p_1, p_2, q_1, q_2$ and the scalars $\alpha^i, \beta^i, \omega, \delta$ are generated in [1,10] to ensure the necessary condition of positivity for $(c_1^i x + d_1^i y + \alpha^i)$ and $(c_2^i x + d_2^i y + \beta^i)$, $i \in \{1, ..., k\}$ as well as $(p^1 x + p^2 y + \omega)$ and $(q^1 x + q^2 y + \delta)$ and the lower level problem is solved using a library IBM CPLEX 12.8 for C++ programs. For each instance (n_1, n_2, m_1, m_2, r) where n_1, n_2, m_1, m_2 and r are respectively the number of leader variables, the number of follower variables, the number of leader constraints, the number of follower constraints and the number of leader objectives, a series of 10 instances of the problem were solved. The obtained performance of the algorithm is summarized in Table 5 where the average CPU time (in seconds) and the average number of efficient solutions are reported. Also, the minimum and maximum values of each measure are reported in brackets.

Table 5 shows that the suggested algorithm can solve small size instances of the problem in a reasonable CPU time. However, the results indicate that the CPU time increases exponentially with the size of the instance. Moreover, the number of objective functions at the upper level does not significantly increase the CPU time. It should also be noted that, for large scale instances, it becomes unrealistic to generate the set of efficient solutions in a reasonable CPU time. This is due to various factors such as the bilevel and multiobjective nature of the problem, the accuracy of the method, the nonlinearity of the objective functions, and the discrete environment of the research domain.

Table 5 Computational results

					Number of efficient solutions	CPU (second)
r	$n_1 + n_2$	$m_1 + m_2$	n_2	m_2	Average [Min; Max]	Average [Min; max]
5	10	10	5	5	7.6[2; 13]	1.65[0.79; 2.82]
	20	10	10	5	18[8; 33]	13.72[11.03; 18.7]
	30	20	10	10	49.7[23;109]	38.30[21.4; 56.23]
	40	30	20	20	35.1[15; 71]	47.06[32.67; 72.66]
	50	40	30	30	28.4[8; 55]	128.91[69.77; 220.98]
	60	50	30	40	39[11; 63]	249.84[216.53; 279.76]
	70	60	30	50	35.5[11; 82]	559.91[494.37; 738.83]
	80	70	30	60	48.3[17; 98]	1113.62[1005.87; 1228.86]
	90	80	50	70	44[22; 125]	2304.6[1972.28; 2526.94]
	100	90	60	80	67.12[34; 146]	4220.07[4019.97; 4419.8]
10	10	10	5	5	8.7[4; 14]	1.83[1.01; 3.34]
	20	10	10	5	36.6[5; 82]	13.1[9.24; 25.37]
	30	20	10	10	153.2[44; 317]	35.88[17.26; 62.69]
	40	30	20	20	129.2[86; 251]	81.75[44.59; 153.84]
	50	40	30	30	114 [27; 354]	190.87[92.83; 297.93]
	60	50	30	40	60.6[17;160]	318.63[243.26; 398.29]
	70	60	30	50	77.7[31; 157]	705.42[520.35; 838.83]
	80	70	30	60	123.9[28; 326]	1375.11[1024.21; 1609.09]
	90	80	50	70	150.75[63; 474]	2917.5[2115.64; 3345.03]
	100	90	60	80	102.62[46; 304]	4833.1[4034.46; 5158.01]

7 Conclusions

This paper proposed an exact algorithm for solving an indefinite quadratic integer bilevel program with multiple objectives at the upper level (1). The branch and bound method, is used to generate integer solutions after solving the upper level problem. When the integer solutions of the upper level are obtained, they are tested for optimality of the lower level problem (5) by using a library IBM CPLEX 12.8 for C++ programs.

Finding the integer optimal bilevel solution to the problem (15) is the focus of cut (17) and the efficient cut [8] is introduced after the initial integer bilevel solution is obtained and new integer efficient solutions are found. The feasible domain is reduced at each step since the efficient cut [8] utilized here eliminates non efficient solutions. When they exist, the $IIQMOBP$ problem's set of efficient solutions is determined after a finite number of iterations.

However, the algorithm can be used with parallel optimization and a more powerful machine to make more it more effective.

Acknowledgments

The Authors work was supported by the Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT) Grant ID: C0656104

Competing interests

The authors declare that they have no competing interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors, the research has been approved by Ethics Committee of the institution responsible for the reaserch.

Consent for publication

Not applicable.

Consent to participate

Not applicable.

Funding

Not applicable.

Availability of data

Not applicable.

Code availability

The method was implemented with the compiler Visual C++ 14.0 and the program was run on an Intel(R) Core (TM) i5 CPU 2.4 GHZ with 4 GB of the main memory and the operating system Windows10 and using a library IBM CPLEX 12.8 for C++ programs

Author's contributions

Fatima Fali desined the algorithm, implemented all experiments, wrote the manuscript and finalised the manuscript.
Mustapha Moulaï revised the manuscript.
All authors read and approved the final manuscript.

References

1. M. Abo-Sinna, A bilevel nonlinear multiobjective decision making under fuzziness, *Journal of Operational Research Society of India*, **38** (2001), 484–495.
2. M.J. Alves, S. Dempe and J. Judice, Computing the Pareto frontier of a bi-objective bi-level linear problem using a multiobjective mixed-integer programming algorithm, *Optimization: A Journal of Mathematical Programming and Operations Research*, **61** (2012), 335–358.
3. Z. Ankhili and A. Mansouri, An exact penalty on bilevel programs with linear vector optimization lower level, *European Journal of Operational Research*, **197** (2009), 36–41.
4. J.F. Bard, Practical Bilevel Optimization. Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 1998.
5. H. Bonnel and J. Morgan, Semivectorial bilevel optimization problem: Penalty approach, *Journal of Optimization Theory and Applications*, **131** (2006), 365–382.
6. H. I. Calvete and C. Gale, Linear bilevel programs with multiple objectives at the upper level, *Journal of Computational and Applied Mathematics*, **234** (2010), 950–959.
7. H. Calvete and C. Gal, Local optimality in quasiconcave bilevel programming, *Monografias del Semin. Matem. Garcia de Galdeano*, **27** (2003), 153–160.
8. M.E-A. Chergui and M. Moulaï, An exact method for a discrete multiobjective linear fractional optimization, *Journal of Applied Mathematics and Decision Sciences*, (2008), 12pp.
9. B. Colson, P. Marcotte and G. Savard, An overview of bilevel optimization, *Annals of Operations Research*, **153** (2007), 235–256.
10. K. Deb and A. Sinha, Solving bilevel multi-objective optimization problems using evolutionary algorithms, *Lecture Notes in Computer Science: Evolutionary Multi-criterion Optimization*, **5467** (2009), 110–124.
11. S. Dempe, Foundations of Bilevel Programming, Kluwer Academic Publishers, Dordrecht, 2002.
12. S. Dempe, Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints, *Optimization*, **52** (2003), 333–359.
13. F. Fali and M. Moulaï, Solving discrete linear fractional bilevel programs with multiple objectives at the upper level, *Journal of Industrial and Management Optimization*, (2022).
14. Y. B. Lv, An exact penalty function approach for solving the linear bilevel multiobjective programming problem, *Filomat*, **29** (2015), 773–779.
15. Y. B. Lv and Z. P. Wan, Solving linear bilevel multiobjective programming problem via exact penalty function approach, *Journal or Inequalities and Applications*, **2015** (2015), 12pp.
16. Y. Lv and Z. Wan, linear bilevel multiobjective optimization problem: penalty approch, *Journal of Industrial and Management Optimization*, **15** (2019), 1213–1223.
17. J.F. Nash, The bargaining problem, *Econometrica*, **18** (1950), 155–162.
18. I. Nishizaki and M. Sakawa,L. Yibing and W. Zhongping, Stackelberg solutions to multiobjective two-level linear programming problem, *Journal of Optimization Theory and Applications*, **103** (1999), 161–182.
19. M. S. Osman, M. A. Abo-Sinna, A. H. Amer and O. E. Emam, A multilevel nonlinear multiob- jective decision making under fuzziness, *Applied Mathematics and Computation*, **153** (2004), 239–252.
20. C.O. Pieume, P. Marcotte, L.P.Fotso and P. Siarry, Generating Efficient Solutions in Bilevel Multi-Objective Programming Problems, *American Journal of Operations Research*, **3** (2013), 289–298.
21. P.G. Saghand, H. Charkhgard and C. Kwon, A Branch-and-Bound Algorithm for a Class of Mixed Integer Linear Maximum Multiplicative Programs: A Multi-objective Optimization Approach, *Computers and Operations Research*, (2018), 1–12.

22. X.Shi and H.Xia, Interactive bilevel multi-objective decision making. *Journal of Operations Research Society*, **48** (1997), 943–949.
23. X.Shi and H.Xia, Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers. *Journal of Multi-Criteria Decision Analysis*, **10** (2001), 27–34.
24. C. Teng, L. Li and H. Li, A class of genetic algorithms on bilevel multiobjective decision making problem, *Journal of Systems Science and Systems Engineering*, **9** (2000), 290–296.
25. R. Narang, S. R. Arora, Indefinite quadratic integer bilevel programming problem with bounded variable, *Journal of operational research society of India (OPSEARCH)*, **4** (2009), 428–448.
26. L. Vicente and P. Calamai, Bilevel and multilevel programming, *Journal of Global Optimization*, **5** (1994), 291–306.
27. Y. Yin, Multiobjective bilevel optimization for transportation planning and management problems, *Journal of advanced Transportation*, **36** (2000), 93–105.
28. N. Maachou and M. Moulaï, Branch and cut method for solving integer indefinite quadratic bilevel programs. *Annals of Operations Research*, (2022).
29. S.R. Arora, R. Arora, Indefinite quadratic bilevel programming problem with multiple objectives at both levels. *International Journal of Optimization*, **1** (2009).
30. Y. Zheng and Z. Wan, A solution method for semivectorial bilevel programming problem via penalty method, *Journal of Applied Mathematics and Computing*, **37** (2011), 207–219.