Pesquisa Operacional (2024) 44: e278191 p.1-30
doi: 10.1590/0101-7438.2023.043.00278191

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

BRANCH-AND-CUT METHOD FOR SOLVING THE INTEGER LINEAR MULTIPLICATIVE BILEVEL PROBLEM

Nacera Maachou[1*] and Mustapha Moulaï[2]

Received September 4, 2023 / Accepted November 14, 2023

ABSTRACT. We study a class of integer bilevel problems, the so-called Integer Linear Multiplicative Bilevel Problem, `ILMBP`, where the constraints are linear and both the upper level problem and the lower level problem are integer linear multiplicative problems. We assume that the *k*-linear factors of the upper level problem and the *l*-linear factors of the lower level problem to be non-negative. In this paper, we propose an algorithm based on branch-and-cut method for solving the Integer Linear Multiplicative Bilevel Problem. First, the continuous upper level linear multiplicative problem is solved, then the optimal integer solution obtained is tested for the optimality of the main problem by solving the lower level linear multiplicative problem. If this solution is not optimal for the problem `ILMBP`, a cut is added to the upper level problem and a new best integer solution is determined. The integer solution is generated by the branching process well known in the branch and bound procedure. Following, the presentation and validation of the algorithm, an example is provided to better visualize the algorithm.

Keywords: bilevel programming, integer programming, linear multiplicative programming, branch and cut, branch and bound.

1 INTRODUCTION

The bilevel programming problem, `BLPP`, is an optimization problem that is constrained by another optimization problem. This mathematical programming model arises when two independent decision makers, ordered within a hierarchical structure, have conflicting objectives. The decision maker at the lower level has to optimize her objective under the given parameters from the upper level decision maker, who, in return, with complete information on the possible reactions of the lower, selects the parameters so as to optimize her own objective. In this sense, the `BLPP` can be perceived as a static Stackelberg game (Simaan & Cruz Jr, 1973) with two independent decision makers.

*Corresponding author
[1]LaROMaD Laboratory, Department of Operations Research, USTHB, Faculty of Mathematics, BP 32, Bab Ezzouar 16111, Algiers, Algeria – E-mail: maachou.nacera.72@gmail.com https://orcid.org/0000-0001-6710-5622
[2]LaROMaD Laboratory, Department of Operations Research, USTHB, Faculty of Mathematics, BP 32, Bab Ezzouar 16111, Algiers, Algeria – E-mail: mmoulai@usthb.dz https://orcid.org/0000-0002-2013-356X

The general form of the bilevel program is:

$$(\mathtt{BP})\ \max_{(x,y)\in S} H(x,y) \tag{1}$$

where, $y \in arg \max_{y\in S(x)} h(x,y)$.
where

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

$$H, h : \mathbb{R}^n \longmapsto \mathbb{R},\ n = n_1 + n_2; \tag{3}$$

$S \subset \mathbb{R}$, defines the common constraint region and

$$S(x) = \{y \in \mathbb{R}^{n_2} : (x,y) \in S\}. \tag{4}$$

Let S_1 be the projection of S onto $\mathbb{R}^{n_1}$. For each $x \in S_1$, the lower level decision maker solves problem (P_1):

$$\mathtt{P}_1 \begin{cases} \max\ h(x,y) \\ \text{s.t} \\ \quad y \in S(x). \end{cases} \tag{5}$$

Let $M(x)$ be the set of optimal solutions to (5):

$$M(x) = \{y^* \in \mathbb{R}^{n_2} : y^* \in arg \max_{y} h(x,y) : y \in S(x)\} \tag{6}$$

The upper level decision maker's feasible set, called the inducible region IR, is implicitly determined by the lower level optimization problem: $\text{IR} = \{(x,y^*) : x \in S_1, y^* \in M(x)\}$ In this paper, we assume that S is not empty. Moreover, the lower level decision maker has some room to react to any decision made by the upper level decision maker, i.e. $M(x) \neq \emptyset$.
In these problems, the upper level is referred to as the leader and the lower level represents the objective of the followers. There are two types of decision makers in a bilevel programming problem `BLPP`. The upper level is referred to as the leader `ULDM` and the lower level represents the objective of the followers `LLDM`. The decision is executed sequentially, from upper level to lower level and each entity makes a decision in a hierarchical manner.

The upper level makes his decision first. Whereas, the lower level may not be satisfied with the decision of `ULDM`. Bilevel programming problem arise in many real world situations, including transportation engineering, organizational design, facility location, production planning, and supply chain management (Wen & Hsu (1991); Wen & Yang (1990); White & Anandalingam (1993)).

As a result, many researchers have conducted in-depth research in this field. The formal formulation of BLPP was proposed in Candler & Townsley (1982); Fortuny-Amat & McCarl (1981); Arora & Gupta (2021, 2018); Neves et al. (2023). In Golpîra (2017), the Karush–Kuhn–Tucker (KKT) conditions are employed to transform the BLPP problem into a single-level, mixed-integer linear programming problem by considering some relaxations. Multichoice optimization has been applied to the bilevel transportation problem in Arora & Gupta (2021), the fuzzy programming approach is employed in order to obtain a satisfactory solution for the decision-makers at the two levels. In Neves et al. (2023), the Dynamic Vehicle Allocation problem is presented and solved using the bilevel programming where the shipper's objective is to minimize shipping delays, while the carrier's objective is to maximize profits.

On the other hand, multiplicative linear programming when there is only one decision level has received a lot of attention in the literature. This attention is due to the fact that these problems arise in many practical applications. For example, the construction of bond portfolios, economic analysis and the design of VLSI chips (Konno & Inori (1989); IRELAND (1971); Maling et al. (1982)). Such optimization problems often arise in game theory. Examples include computing the Nash solution to a bargaining problem (Nash Jr (1950)) and computing an equilibrium of a linear Fisher or a Kelly capacity allocation market (Chakrabarty et al. (2006); Jain & Vazirani (2007); Vazirani (2012a,b)).

Solving a bilevel (more generally, hierarchical) optimization problem, even in its simplest form, is a difficult task. Many methods have been proposed and discussed by various authors Dempe et al. (2014); Dempe & Dutta (2012); Quynh et al. (2012); Vicente & Calamai (1994); Maachou & Moulaï (2015) with the aim of developing various ways to reduce original bilevel programming problems to equivalent single-level ones, thus making them easier to solve using mathematical programming software packages.

Integer bilevel programming has been addressed in Thirwani & Arora (1998); Narang & Arora (2009); Maachou & Moulaï (2022). Thirwani & Arora (1998) developed an algorithm for solving `QBLPP` for integer variables. They solved the problem by linearization technique and obtained an integer solution of the `QBLPP` by using Gomory cut and dual simplex method. Narang & Arora (2009) presented an algorithm for solving an indefinite integer `QBLPP` with bounded variables. They solved the problem by solving the relaxed problem and developed a mixed integer cut solution technique for finding the integer solution. Maachou & Moulaï (2022) developed a new algorithm based on the branch and cut method to solve the integer indefinite quadratic bilevel programs where the upper level problem and the lower level problem are integer indefinite quadratic programs. The problem is solved in its original formulation where the decision maker of the upper level chooses his strategy first. Then, using the optimal solution of the upper level problem, the decision maker of the lower level has to select his best strategy optimizing his own objective. The optimal solution of the integer indefinite quadratic problem belongs to the efficient solutions set of the corresponding bicriteria problem.

In this paper, we consider a special class of nonlinear bilevel programming in which the upper-level's objective function is a product of k-linear functions and the lower-level's objective function is a product of l-linear functions. This class of problems is refered as `ILMBP`. Assuming that the k-linear factors of the upper-level's objective function and l-linear factors of the lower-level's objective function are positive, maximization cases are considered for the two levels.

The algorithm developed here for solving ILMBP is based on a branch and cut technique, such that, to get the integer optimal solution of our main problem, the upper level is solved, and if its solution is non optimal for ILMBP, a cut is added to the initial domain of the upper level to avoid the integer solution, and a new integer solution is found. The branch and bound process, allows us to determine the integer solution.

In the following Section, the Integer Linear Multiplicative Bilevel Problem ILMBP is defined and formulated. In Section 3, an algorithm is presented for solving the k-linear multiplicative problem. A branch and cut method is presented in section 4 for solving the Integer Linear Multiplicative Bilevel Problem ILMBP and a more detailed description of the algorithm for the whole problem is given. In section 5, a numerical example is presented for better understanding the algorithm of the proposed method and in section 6 computational results are presented. Section 7, concludes the paper.

2 DEFINITIONS AND NOTATIONS

Let $\tilde{S}$ be the set of feasible solutions $(x,y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, not empty and bounded, satisfying the constraints

$$\begin{cases} A^1x + A^2y \leq b \\ x \geq 0, y \geq 0 \end{cases} \tag{7}$$

where A^1 is an $m \times n_1$ matrix, A^2 is an $m \times n_2$ matrix and b is a vector of $\mathbb{R}^m$.

Let c_1^i be a row vector of $\mathbb{R}^{n_1}$, c_2^i be a row vector of $\mathbb{R}^{n_2}$ for $i = \overline{1,k}$, p_1^j be a row vector of $\mathbb{R}^{n_1}$, p_2^j be a row vector of $\mathbb{R}^{n_2}$ for $j = \overline{1,l}$, the constants α_i for $i = \overline{1,k}, \beta_j$ for $j = \overline{1,l}$ are real and $\mathbb{Z}^n$ is the set of integer numbers. the Integer Linear Multiplicative Bilevel Problem ILMBP, intended to be studied, can be mathematically stated as:

$$\texttt{ILMBP} \begin{cases} \max\limits_{(x,y)} F(x,y) = \prod\limits_{i=1}^{k}(c_1^i x + c_2^i y + \alpha_i) \\ \text{where } y \text{ solves} \\ \quad \begin{cases} \max\limits_{y} f(x,y) = \prod\limits_{j=1}^{l}(p_1^j x + p_2^j y + \beta_j) \\ s.t \\ \quad (x,y) \in \mathscr{D} = \tilde{S} \cap \mathbb{Z}^n \end{cases} \end{cases} \tag{8}$$

The decision vector x is controlled by the Upper Level Integer Linear Multiplicative Problem ULILMP and the decision vector y is controlled by the Lower Level Integer Linear Multiplicative Problem LLILMP.

Mathematically, the upper level problem ULILMP can be written as:

$$\texttt{ULILMP} \begin{cases} \max\limits_{(x,y)} F(x,y) = \prod\limits_{i=1}^{k}(c_1^i x + c_2^i y + \alpha_i) \\ s.t \quad (x,y) \in \mathscr{D} \end{cases} \tag{9}$$

and the lower level problem LLILMP can be written as:

$$\text{LLILMP}\begin{cases}\max\limits_{y}\ f(x,y)=\prod\limits_{j=1}^{l}(p_1^j x+p_2^j y+\beta_j)\\ s.t\quad (x,y)\in\mathscr{D}\end{cases}\tag{10}$$

Let us denote the feasible region of ULILMP $S=\{(x,y)\in\mathbb{R}^{n_1}\times\mathbb{R}^{n_2}\,|\,x,y\in\tilde{S},(c_1^i x+c_2^i y+\alpha_i)\geq 0,i=\overline{1,k}\}$ and $z=(x,y)$,
$c^i=(c_1^i,c_2^i)^T,i=\overline{1,k}$, the upper level problem ULILMP can be rewritten as:

$$\text{ULILMP}\begin{cases}\max\limits_{(x,y)}\ F(x,y)=\prod\limits_{i=1}^{k}(c^i z+\alpha_i)\\ s.t\quad (x,y)\in\mathscr{D}_1=S\cap\mathbb{Z}^n\end{cases}\tag{11}$$

and the feasible region of LLILMP $\hat{S}=\{(x,y)\in\mathbb{R}^{n_1}\times\mathbb{R}^{n_2}\,|\,x,y\in\tilde{S},(p_1^j x+p_2^j y+\beta_j)\geq 0,j=\overline{1,l}\}$ and $z=(x,y)$,
$p^j=(p_1^j,p_2^j)^T,j=\overline{1,l}$ the lower level problem LLILMP can be rewritten as:

$$\text{LLILMP}\begin{cases}\max\limits_{y}\ f(x,y)=\prod\limits_{j=1}^{l}(p^j z+\beta_j)\\ s.t\quad (x,y)\in\mathscr{D}_2=\hat{S}\cap\mathbb{Z}^n\end{cases}\tag{12}$$

The main problem ILMBP is resolved in this paper in its original form. First, the decision maker of the upper level selects his strategy and determines his optimal solution x^*. After employing it, the decision maker of the lower level has to choose his best strategy y^* optimizing his own objective (Dempe (2002)). In the following section, we will discuss how to solve the integer linear multiplicative problem.

3 PROCEDURE TO SOLVE THE INTEGER LINEAR MULTIPLICATIVE PROBLEMS

Consider the following integer linear multiplicative problem:

$$\text{ILMP}\begin{cases}\max\ g(z)=\prod\limits_{i=1}^{k}(u^i z+\alpha_i)\\ s.t\quad z\in\mathscr{D}\end{cases}\tag{13}$$

where, $z=(x,y)\in\mathbb{R}^{n_1+n_2}$, $\mathscr{D}=\mathscr{X}\cap\mathbb{Z}^n$, $\mathscr{X}=\{z\in\mathbb{R}^{n_1+n_2}\,|\,Az\leq b,\quad z\geq 0,(u^i z+\alpha_i)\geq 0,i=\overline{1,k}\}$ is a non empty bounded set, $b\in\mathbb{R}^m$, u^i are row vectors of $\mathbb{R}^{n_1+n_2}$, α_i are real, $i=\overline{1,k}$ and A is an $m\times(n_1+n_2)$ matrix.

Given the following multicriteria linear problem associated to `ILMP`

$$\texttt{IMLP}\begin{cases} \max\ g_1(z) = u^1 z + \alpha_1 \\ \max\ g_2(z) = u^2 z + \alpha_2 \\ . \\ . \\ . \\ \max\ g_k(z) = u^k z + \alpha_k \\ s.t \quad z \in \mathscr{D} \end{cases} \tag{14}$$

Aneja et al. (1984) showed that in a two-factor continuous multiplicative problem $(k = 2)$, the optimal solution of (13) is an efficient solution of problem (14). Maachou & Moulaï (2022) have adapted this result to the case of an integer multiplicative problem. In what follows, we will extend this result to the case where $k \geq 2$.

Theorem 1. *An optimal solution $\hat{z}$ of problem `ILMP` is an efficient solution of integer multicriteria linear problem `IMLP`.*

Proof. Suppose that $\hat{z}$ is an optimal solution of problem `ILMP` but it is not an efficient solution of Integer Multicriteria Linear Problem `IMLP`.
By definition, this implies that there must exist a feasible solution denoted by $z \in \mathscr{D}$ that dominates $\hat{z}$. In other terms,
for $g_i(z) = g_i$ and $g_i(\hat{z}) = \hat{g}_i, i = \overline{1,k}$ we must have that either

$\hat{g}_i \leq g_i$, for all $i \in \{1,...,k\}$ and $\hat{g}_i < g_i$ for at least one $i \in \{1,...,k\}$ Also, by assumptions of problem `ILMP` we know that $\hat{g}_i > 0, i = \overline{1,k}$.
Therefore, we must have that

$$0 < \prod_{i=1}^{k} \hat{g}_i < \prod_{i=1}^{k} g_i$$

Therefore, $\hat{z}$ cannot be an optimal solution. □

Thus, an Integer Linear Multiplicative Problem `ILMP` is equivalent to the following problem (Nash Jr (1950)):

$$\texttt{ILMP'}\begin{cases} \max\ g(z) = \prod_{i=1}^{k} g_i(z) \\ s.t \quad z \in \mathscr{D}_E \end{cases} \tag{15}$$

where $g_i(z) = (u^i z + \alpha_i), \quad i = \overline{1,k}$ and $\mathscr{D}_E$ is the set of efficient solutions of the Integer Multicriteria Linear Problem `IMLP`.

To solve the Integer Linear Multiplicative Problem `ILMP`, one would first have to generate the whole set $\mathscr{D}_E$, the set of efficient solutions of the Integer Multicriteria Linear Problem `IMLP`, and then choose the best optimal value of g.

Many methods have been presented to determine the efficient set of IMLP using the concept of efficiency.

Definition 1. *A solution z is dominated by the other solution z^*, if $g_i(z^*) \geq g_i(z), for\ i = \{1,...,k\}$ and $g_i(z^*) > g_i(z), for\ at\ least\ one\ i = \{1,...,k\}$.*

Definition 2. *A feasible solution z is called an efficient solution for* IMLP *problem if and only if there is no other $\tilde{z} \in \mathscr{D}_E$, such that $g_i(\tilde{z}) \geq g_i(z), for\ i = \{1,...,k\}$ and $g_i(\tilde{z}) > g_i(z), for\ at\ least\ one\ i = \{1,...,k\}$. Otherwise, z is dominated by $\tilde{z}$.* In the next section, we present in detail the method we used to generate the efficient set of IMLP.

3.1 Method for finding the integer efficient solutions multicriteria linear problem IMLP

To generate the efficient set of IMLP, we use a strategy based on the branch and cut method, which is described as follows:

3.1.1 Description of the method

Let $\mathscr{D}_E$, $\bar{\mathscr{D}}_E$ be the sets of all efficient solutions and dominated solutions, respectively, of the integer multicriteria linear problem (14) initialized to the empty set, $l = 0$ and $\mathscr{X}_0 = \mathscr{X}$.
The search for the efficient set is an implicit exploration structured as a tree. At each node l of the tree, we optimize one of k functions $g_i, i \in \{1,...,k\}$ on the subdomain $\mathscr{X}_l$ (the choice of function will not change the final efficient set, but may change the order of the elements in it. In the following, we choose the first function). The subproblem associated with a node l is defined as follows:

$$\mathrm{P}_l \begin{cases} \max\ g_1(z) = u^1 z + \alpha_1 \\ s.t \quad z \in \mathscr{X}_l \end{cases} \tag{16}$$

where, $z = (x,y) \in \mathbb{R}^{n_1+n_2}$. In our method, the problem P_l is solved using the simplex method (eventually dual simplex method). Three situations may arise during problem-solving:

1. The problem $\mathbf{P}_l$ **has no solution:** The node has no descendant and it is fathomed.

2. The problem $\mathbf{P}_l$ **has non integer solution:** Let $\hat{z}_j^{(l)}$ be one component of the non integer optimal solution $\hat{z}^{(l)}$ such that $\hat{z}_j^{(l)} = \nu_j$ where ν_j is a fractional number. The node l of the tree is then separated into two nodes, l_1 and l_2, $(l_1 > l, l_2 > l, l_1 \neq l_2)$, which partition the feasible set $\mathscr{X}_l$ into two parts ($\mathscr{X}_{l_1}$ and $\mathscr{X}_{l_2}$) by adding to $\mathscr{X}_l$ new branching constraints $\hat{z}_j^{(l)} \leq \lfloor \nu_j \rfloor$ and $\hat{z}_j^{(l)} \geq \lfloor \nu_j \rfloor + 1$, respectively.

3. The problem $\mathbf{P}_l$ **has an integer solution:** Let $\hat{z}^{(l)}$ the optimal solution of P_l.

   - **We update the sets $\mathscr{D}_E$, $\bar{\mathscr{D}}_E$.** The integer optimal solution $\hat{z}^{(l)}$ is then compared to the solutions of the set $\mathscr{D}_E$. Thus, the sets $\mathscr{D}_E$ and $\bar{\mathscr{D}}_E$ are updated as follows: At the node l, corresponding to the solution $\hat{z}^{(l)}$, if there exists $y \in \mathscr{D}_E$ such that

$(g_1(y),...,g_k(y))$ dominates $(g_1(\hat{z}^{(l)}),...,g_k(\hat{z}^{(l)}))$, then $\mathscr{D}_E$ is not updated and $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E \cup \{\hat{z}^{(l)}\}$. Otherwise, if there is no solution $y \in \mathscr{D}_E$ such that $(g_1(y),...,g_k(y))$ dominates $(g_1(\hat{z}^{(l)}),...,g_k(\hat{z}^{(l)}))$, then the solution $\hat{z}^{(l)}$ is added to the set $\mathscr{D}_E$ and all solutions where $y \in \mathscr{D}_E$ and $(g_1(\hat{z}^{(l)}),...,g_k(\hat{z}^{(l)}))$ dominates $(g_1(y),...,g_k(y))$ are removed, $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E \cup \{y\}$.

- **The cutting plane:** At node l, let I_l, N_l be the index sets of basic and nonbasic variables respectively (for $j \in N_l, z_j = 0$) of the integer optimal solution $\hat{z}^{(l)}$ at the optimal simplex table. We define the constraint:

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

where, $H_l = \left\{ j \in N_l \mid \exists i \in \{1,...,k\}, \bar{u}^i_j > 0 \right\} \cup \left\{ j \in N_l \mid \bar{u}^i_j = 0, \forall i \in \{1,...,k\} \right\}$ and $\bar{u}^i_j$ is the j^{th} component of the reduced gradient vector $\bar{u}^i$ for each objective function g_i, $i \in \{1,...,k\}$ at the optimal simplex table.

The inequality (17) is called efficient cut (Chergui & Moulaï (2008)).

If $H_l = \emptyset$ then the remaining domain contains no efficient solution and the node l is fathomed (Proposition 1). Conversely, if $H_l \neq \emptyset$, then the node l has a successor $l+1$. The corresponding domain of the node $l+1$, the successor of l, is given by applying (17) to $\hat{z}^{(l)}$:

$$\mathscr{X}_{l+1} = \left\{ z \in \mathscr{X}_l : \sum_{j \in H_l} z_j \geq 1 \right\} \tag{18}$$

Each node l of the tree is fathomed if the corresponding problem $\mathtt{P}_l$ is unfeasible or $H_l = \emptyset$.
The algorithm can be summarized as follows:

Algorithm 1 Generating the efficient set of `IMLP`

Step 1: **Initialization** $l = 0$, $\mathscr{X}_0 = \mathscr{X}$, $\mathscr{D}_E = \emptyset$, $\bar{\mathscr{D}}_E = \emptyset$.

Step 2: **General Step**.

While there is a non-fathomed node in the tree, choose the node l not yet fathomed and solve the corresponding linear problem (16).

- If the problem (16) is unfeasible, then the corresponding node l is fathomed.
- Otherwise, let $(\hat{z}^{(l)})$ be an optimal solution of the problem (16).
  1. If $\hat{z}^{(l)}$ is integer, goto Step 2a.
  2. Otherwise, goto Step 2b.

Step 2a:

- If the vector $(g_1(\hat{z}^{(l)}),...,g_k(\hat{z}^{(l)}))$ is not dominated by the vector $(g_1(z),...,g_k(z))$ for any $z \in \mathscr{D}_E$ then $\mathscr{D}_E = \mathscr{D}_E \cup \{\hat{z}^{(l)}\}$ and $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E$.
- If there is a solution $z \in \mathscr{D}_E$ such that $(g_1(\hat{z}^{(l)}),...,g_k(\hat{z}^{(l)}))$ dominates $(g_1(z),...;g_k(z))$ then $\mathscr{D}_E = \mathscr{D}_E \backslash \{z\} \cup \{\hat{z}^{(l)}\}$ and $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E \cup \{z\}$.

Determine sets N_l, H_l;

1. If $H_l = \emptyset$ then the node l is fathomed and goto Step 2;
2. Otherwise, add the efficient cut $\sum_{j \in H_l} z_j \geq 1$ to problem (16) and goto Step 2, a new node is created, $l = l + 1$.

Step 2b: Branching Process Since there is at least one non-integer value among the values $\hat{z}^{(l)}$, choose one of them such that $\hat{z_j}^{(l)} = \nu_j$, where ν_j is a fraction. Partition the feasible set $\mathscr{X}_l$ into two parts ($\mathscr{X}_{l_1}$ and $\mathscr{X}_{l_2}$) $(l_1 > l+1, l_2 > l+1, l_1 \neq l_2)$ by adding to $\mathscr{X}_l$ new branching constraints $\hat{z_j}^{(l)} \leq \lfloor \nu_j \rfloor$ and $\hat{z_j}^{(l)} \geq \lfloor \nu_j \rfloor + 1$, respectively, and goto Step 2.

3.2 Theoretical results

The following theoretical marks show that the Algorithm 1 determines the set of integer efficient solutions to problem (14) in a finite number of iterations.

Theorem 2. *Assume that $H_l \neq \emptyset$ at the current integer solution $\hat{z}^{(l)}$.*
If $z \neq \hat{z}^{(l)}$ is an integer efficient solution of problem (14) in domain $\mathscr{X}_l$, then $z \in \mathscr{X}_{l+1}$ (the node $l+1$ is the successor of node l).

Proof. Let $z \neq \hat{z}^{(l)}$ be an integer solution in domain $\mathscr{X}_l$ such that $z \notin \mathscr{X}_{l+1}$, this implies

$$z \in \left\{ z \in \mathscr{X}_l : \sum_{j \in H_l} z_j < 1 \right\} \tag{19}$$

Therefore, z satisfy the inequalities

$$\sum_{j \in H_l} z_j < 1 \tag{20}$$

$$\sum_{j \in N_l \setminus H_l} z_j \geq 1 \tag{21}$$

This means that $z_j = 0$ for all $j \in \mathscr{X}_l$, and $z_j \geq 1$ for at least one index $j \in N_l \setminus H_l$. From the simplex table corresponding to the solution $\hat{z}^{(l)}$ we deduce the following equality for all criteria $i \in \{1, ..., k\}$:

$$u^i z = u^i \hat{z}^{(l)} + \sum_{j \in N_l} \bar{u_j}^i z_j = u^i \hat{z}^{(l)} + \sum_{j \in H_l} \bar{u_j}^i z_j + \sum_{j \in N_l \setminus H_l} \bar{u_j}^i z_j \tag{22}$$

Then,

$$u^i z = u^i \hat{z}^{(l)} + \sum_{j \in N_l \setminus H_l} \bar{u_j}^i z_j \tag{23}$$

Thus, $u^i z \leq u^i \hat{z}^{(l)}$ for all criteria $i \in \{1, ..., k\}$, with $u^i z < u^i \hat{z}^{(l)}$ for at least one criterion since $\overline{u}_j^i \leq 0$ for all $j \in N_l \setminus H_l$.

Hence, the criterion vector $(g^1(z), ..., g^k(z))$ is dominated by $(g^1(z^{*(l)}), ..., g^k(z^{*(l)}))$ and z is not efficient. $\square$

Proposition 1. *Suppose that $H_l = \emptyset$ at the current integer solution $\hat{z}^{(l)}$, then there are no efficient solutions in the remaining domain $\mathscr{X}_l \setminus \hat{z}^{(l)}$.*

Proof. Assume $H_l = \emptyset$, then $z^{*(l)}$ is an optimal integer solution for all criteria. Therefore, $z^{*(l)}$ is an ideal point in the domain $\mathscr{X}_l$, and $\mathscr{X}_l \setminus \{z^{*(l)}\}$ does not contain efficient solutions. $\square$

Theorem 3. *The proposed algorithm generates all efficient integer solutions of the integer multicriteria linear problem (14) in a finite number of iterations, if such solutions exist.*

Proof. Let $\mathscr{X}$ be a finite bounded set containing $\mathscr{D}$, the set of integer feasible solutions of the `IMLP` problem. Since there are only finitely many integer solutions, the efficient set $\mathscr{D}_E$ has finite cardinality. As a result, the search tree will have a fixed number of branches. Consequently, the algorithm ends after a finite number of steps. $\square$

3.3 An algorithm to solve the Integer Linear Multiplicative Problems ILMP

Using Theorem 1, we propose the following algorithm (Algorithm 2) to solve the `ILMP` problem (15).

Algorithm 2 Solving Integer Linear Multiplication Problems `ILMP`

Step 0: **Initialization** From `ILMP`, we determine the corresponding integer multicriteria problem `IMLP`.

Step 1: Using the above algorithm (Algorithm 1), we generate the set of all integer efficient solutions of the `IMLP` problem .

Step 2: Let $\mathscr{D}_E$ be the set of integer efficient solutions of `IMLP` then, the integer optimal solution of the problem `ILMP` is $z^* = \arg\max_{z \in \mathscr{D}_E} g(z)$ and $g_{opt} = g(z^*)$.

4 A METHOD TO SOLVE THE INTEGER LINEAR MULTIPLICATIVE BILEVEL PROBLEMS ILMBP

An exact method based on the branch and cut technique is given in depth in this section. The convergence of the method is then demonstrated.

4.1 Description of the method

The following general steps solve the Integer Linear Multiplicative Bilevel Problem `ILMBP`:

- **Step 1:** In first, we initialize $F_{opt} = -\infty$, $f_{opt} = -\infty$, $z_{opt} = \emptyset$, $p = 0$ and $S_0 = S$.
- **Step 2:** Using the Algorithm 2, we solve the upper level problem (24) over the subdomain S_p corresponding to the node p.

$$\mathtt{ULILMP} \begin{cases} \max\limits_{(z)} \; F(z) = \prod\limits_{i=1}^{k} (c^i z + \alpha_i) \\ s.t \quad z \in S_p \end{cases} \tag{24}$$

  If the problem (24) has no solution, the main problem is unfeasible and the algorithm stops. If not, proceed to step 3.

- **Step 3:** Let $z^{*(p)} = (x^{*(p)}, y^{*(p)})$ be the integer optimal solution of the upper level and $F_{opt} = F(z^{*(p)})$. I_p, N_p are the index sets of the basic and nonbasic variables, respectively, of the optimal solution $z^{*(p)}$ obtained by Algorithm 2. This solution is then tested for the optimality of the main problem by solving the lower level problem (25) (Dempe (2002)): We fix $x = x^{*(p)}$ and use again the Algorithm 2 described above to solve the lower level problem (25),

$$\mathtt{LLILMP} \begin{cases} \max\limits_{y} \; f(x,y) = \prod\limits_{j=1}^{l} (p_1^j x^{*(p)} + p_2^j y + \beta_j) \\ s.t \quad (x^{*(p)}, y) \in \hat{S}_p \end{cases} \tag{25}$$

  The integer optimal solution is $\tilde{y}$. There are two possible scenarios:

1. If $y^{*(p)} = \tilde{y}$, then $z^{*(p)} = (x^{*(p)}, y^{*(p)})$ is the reached integer optimal solution of the main problem (8), $F_{opt} = F(z^{*(p)})$, $f_{opt} = f(z^{*(p)})$, $z_{opt} = z^{*(p)}$ and the algorithm stops.
2. Otherwise, if $y^{*(p)} \neq \tilde{y}$ or the lower level (29) is unfeasible, go to step 4.

- **Step 4:** Instead of initializing $\mathscr{D}_E = \emptyset$ and $\bar{\mathscr{D}}_E = \emptyset$ in Algorithm 2, we initialize $\mathscr{D}_E = \mathscr{D}_E^{p-1}$ and $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E^{p-1}$(for $p \geq 1$) in order to avoid re-searching previously discovered solutions. We then proceed to update the sets $\mathscr{D}_E^p$ and $\bar{\mathscr{D}}_E^p$.
  - $\mathscr{D}_E^p = \mathscr{D}_E^p \backslash \{z^{*(p)}\}$ and $\bar{\mathscr{D}}_E^p = \bar{\mathscr{D}}_E^p$.
    - If there is a solution $\bar{z} \in \bar{\mathscr{D}}_E^p$ such that the vector $(F_1(\bar{z}), \ldots, F_k(\bar{z}))$ is not dominated by the vector $(F_1(z), \ldots, F_k(z))$ for any $z \in \mathscr{D}_E^p$, then $\mathscr{D}_E^p = \mathscr{D}_E^p \cup \{\bar{z}\}$, $\bar{\mathscr{D}}_E^p = \bar{\mathscr{D}}_E^p \backslash \{\bar{z}\}$, goto step 5.

- **Step 5:** The solution $z^{*(p)}$ in this step is not optimal for the main problem `ILMBP`, we want to eliminate this solution from the domain S_p and continue the exploration from node p, where S_{p+1} is the corresponding domain of node $p+1$, the successor of p, obtained by applying the inequality (27) to $z^{*(p)}$:

$$S_{p+1} = \left\{ z \in S_p : \sum_{j \in N_p} z_j \geq 1 \right\} \tag{26}$$

The inequality (27) is defined as follows:

$$\sum_{j \in N_p} z_j \geq 1. \tag{27}$$

where N_p is the set of indices of nonbasic variables of the optimal solution $z^{*(p)}$ obtained by Algorithm 2. The integer solution $z^{*(p)}$ of the upper problem (24) is eliminated by the inequality (27), which is known as a Dantzig cut, and it is added to the successor nodes of p.

$\mathscr{D}_E = \mathscr{D}_E^p$, $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E^p$, $p = p+1$, and goto step 2.

During each iteration of Algorithm 3, the upper level is solved using Algorithm 2 (which uses Algorithm 1). The obtained solution is then tested for optimality of the main problem. If this solution is not optimal, we remove it from the domain and continue the exploration. The algorithm stops if the optimal solution for the main problem is obtained or if there is no solution at the upper level.

The algorithm used to obtain an integer optimal solution to our main problem (8) is can be summarized as follows:

Algorithm 3 An algorithm to solve the `ILMBP` problem

Step 1: **Initialization** $p = 0$, $F_{opt} = -\infty$ and $f_{opt} = -\infty$, $z_{opt} = \emptyset$.

Solve the upper level problem $\texttt{ULILMP}_0$. Using the above Algorithm 2, the sets of efficient and dominated solutions $\mathscr{D}_E$ and $\bar{\mathscr{D}}_E$ respectively are determined,

$\mathscr{D}_E^0 = \mathscr{D}_E$, $\bar{\mathscr{D}}_E^0 = \bar{\mathscr{D}}_E$.

- If $\texttt{ULILMP}_0$ is unfeasible, then the bilevel integer optimal solution of the main problem (8) does not exist, and the algorithm stops.
- Otherwise, let $z^{*(0)} = (x^{*(0)}, y^{*(0)})$, the optimal solution of the upper level $(ULILMP_0)$, $F_{opt} = F(z^{*(0)})$ and goto step 3.

Step 2: **General Step.** Use the above Algorithm 2 to solve the following upper level problem at node p.

$$\texttt{ULILMP}_p \begin{cases} \max\limits_{(z)} \ F(z) = \prod_{i=1}^{k}(c^i z + \alpha_i) \\ s.t \quad z \in S_p \end{cases} \tag{28}$$

The sets of efficient and dominated solutions, $\mathscr{D}_E$ and $\bar{\mathscr{D}}_E$ respectively, are then updated, $\mathscr{D}_E^p = \mathscr{D}_E$, $\bar{\mathscr{D}}_E^p = \bar{\mathscr{D}}_E$. Let $z^{*(p)} = (x^{*(p)}, y^{*(p)})$ be the optimal solution of the upper level $\texttt{ULILMP}_p$.

- If $\texttt{ULILMP}_p$ is unfeasible or $F(z^{*(p)}) > F_{opt}$ (for $p \geq 1$), then the corresponding node p is fathomed.
- Otherwise, $F_{opt} = F(z^{*(p)})$ and goto step 3.

Step 3: Put $x = x^{*(p)}$ and solve the lower level problem

$$\texttt{LLILMP} \begin{cases} \max\limits_{y} \ f(x,y) = \prod_{j=1}^{l}(p_1^j x^{*(p)} + p_2^j y + \beta_j) \\ s.t \quad (x^{*(p)}, y) \in \hat{S}_p \end{cases} \tag{29}$$

using the same technique as in Algorithm 2 above.

- Let $\tilde{y}$ be the integer optimal solution of the lower level problem (29).
  1. If $y^{*(p)} = \tilde{y}$ then $z^{*(p)} = (x^{*(p)}, y^{*(p)})$ is the bilevel integer optimal solution of the main problem (8), $F_{opt} = F(z^{*(p)})$, $f_{opt} = f(z^{*(p)})$, $z_{opt} = z^{*(p)}$ and the algorithm stops.
  2. Otherwise, goto step 4.

Step 4: Update the set $\mathscr{D}_E^p$ and the set $\bar{\mathscr{D}}_E^p$. $\mathscr{D}_E^p = \mathscr{D}_E^p \backslash \{z^{*(p)}\}$ and $\bar{\mathscr{D}}_E^p = \bar{\mathscr{D}}_E^p$.

- If there is a solution $\bar{z} \in \bar{\mathscr{D}}_E^p$ such that the vector $(F_1(\bar{z}), ..., F_k(\bar{z}))$ is not dominated by the vector $(F_1(z), ..., F_k(z))$ for any $z \in \mathscr{D}_E^p$, then $\mathscr{D}_E^p = \mathscr{D}_E^p \cup \{\bar{z}\}$, $\bar{\mathscr{D}}_E^p = \bar{\mathscr{D}}_E^p \backslash \{\bar{z}\}$, goto step 5.

Step 5: Determine the set N_p corresponding to the integer solution $z^{*(p)}$ obtained by Algorithm 2. A Dantzig cut (27) is added to the corresponding domain to the integer solution $z^{*(p)}$. $\mathscr{D}_E = \mathscr{D}_E^p$, $\bar{\mathscr{D}}_E = \bar{\mathscr{D}}_E^p$, $p = p + 1$, and goto step 2.

Theorem 4. *The proposed algorithm generates an optimal integer solution of the program* `ILMBP` *in a finite number of iterations, if such a solution exists.*

Proof. Let $\mathscr{D}$ be the set of integer feasible solutions of the `ILMBP` problem, be a finite bounded set contained in the set of the feasible solutions $\tilde{S}$. The set $\mathscr{D}$ has finite cardinality because it contains finite integer solutions. Thus, at each step p of Algorithm 3, if an integer solution $z^*(p)$ of the main problem is not optimal, the Dantzig cut (27) is constructed and eliminates $z^*(p)$, then a new node is determined. Therefore, when the set S_p does not contain an integer optimal solution, the subtree rooted at node p is explored. Eventually, after a finite number of steps, all nodes are fathomed, and the search tree has a finite number of branches. As a result, the algorithm concludes within a finite number of steps. □

5 A DIDACTIC EXAMPLE

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

$$
(\texttt{ILMBP})\left\{\begin{array}{l}
\max\limits_{x_1}\ F(x_1,y_1,y_2) = (x_1+2y_2+3)(3y_1+2)(2x_1+y_1+2)(y_2+1)\\
\text{where } (y_1,y_2) \text{ solves}\\
\qquad\qquad\left\{\begin{array}{ll}
\max f(x_1,y_1,y_2) = (y_1+1)(x_1+y_1-y_2+3) & \\
s.t & \\
3x_1+y_1+2y_2 & \leq 5\\
y_1+y_2 & \leq 3\\
x_1+2y_1+y_2 & \leq 2\\
3y_1+2y_2 & \leq 6\\
x_1\geq 0, y_1\geq 0, y_2\geq 0, integers &
\end{array}\right.
\end{array}\right. \tag{30}
$$

To solve the `ILMBP`, we use Algorithm 3.

Initialization $p=0$, $F_{opt}=-\infty$ and $f_{opt}=-\infty$, $z_{opt}=\emptyset$.
Solve the upper level problem $\texttt{ULILMP}_0$, by using Algorithm 2, the set of efficient and dominated solutions, respectively, $\mathscr{D}_E$ and $\bar{\mathscr{D}}_E$ are determined, $D_E^0 = D_E$, $\bar{D}_E^0 = \bar{D}_E$.
For $(y_1,y_2)=(x_2,x_3)$, the Upper Level Integer Linear Multiplicative $\texttt{ULILMP}_0$ can be written as:

$$
(\texttt{ULILMP}_0)\left\{\begin{array}{ll}
\max\ F(x_1,x_2,x_3) = (x_1+2x_3+3)(3x_2+2)(2x_1+x_2+2)(x_3+1) & \\
s.t & \\
3x_1+x_2+2x_3 & \leq 5\\
x_2+x_3 & \leq 3\\
x_1+2x_2+x_3 & \leq 2\\
3x_2+2x_3 & \leq 6\\
x_1\geq 0, x_2\geq 0, x_3\geq 0, integers &
\end{array}\right. \tag{31}
$$

To solve the upper level problem $\mathtt{ULILMP}_0$, consider the corresponding Integer Multicriteria Linear Problem $\mathtt{IMLP}$:

$$(\mathtt{IMLP})\begin{cases} \max\ g_1(x_1,x_2,x_3) = (x_1+2x_3+3) \\ \max\ g_2(x_1,x_2,x_3) = (3x_2+2) \\ \max\ g_3(x_1,x_2,x_3) = (2x_1+x_2+2) \\ \max\ g_4(x_1,x_2,x_3) = (x_3+1) \\ s.t \\ 3x_1+x_2+2x_3 \quad \leq 5 \\ x_2+x_3 \quad \leq 3 \\ x_1+2x_2+x_3 \quad \leq 2 \\ 3x_2+2x_3 \quad \leq 6 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, integers \end{cases} \tag{32}$$

For $l=0$, $\mathscr{X}_0 = \mathscr{X}$, $D_E = \emptyset$, $\bar{D}_E = \emptyset$, $F_{opt} = -\infty$, $f_{opt} = -\infty$ and $z_{opt} = \emptyset$. The program $\mathtt{P}_0$ is solved. The integer optimal solution $z^{*(0)} = (0,0,2)$, is given in Table 1 .

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

The criterion vector is $(7,2,2,3)$, $D_E = \{(0,0,2)\}$, $\bar{D}_E = \emptyset$.

$H_1 = \{1,2\}$. The efficient cut $x_1+x_2 \geq 1$ is added to $\mathtt{P}_0$ constraints. We obtain $\mathtt{P}_1$, solve the problem, and the integer optimal solution $z^{*(1)} = (1,0,1)$ is given in Table 2.

The criterion vector is $(6,2,4,2)$. Since this solution is efficient, we have $D_E = \{(0,0,2),(1,0,1)\}$, $\bar{\mathscr{D}}_E = \emptyset$. $H_2 = \{2,8\}$, apply the efficient cut $x_2+x_8 \geq 1$, and solve the new problem $\mathtt{P}_2$, the optimal solution $z^{*(2)} = (8/5,1/5,0)$ is given by Table 3, which is not an integer.

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 2.

B_2	x_4	x_6	x_9	*Rhs*
x_2	-1/5	2/5	-1/5	1/5
x_5	1/5	-7/5	-4/5	14/5
x_3	0	1	1	0
x_7	3/5	-16/5	-7/5	27/5
x_1	2/5	-4/5	-3/5	8/5
x_8	1/5	-2/5	-4/5	4/5
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-2/5	-6/5	-7/5	-23/5
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	3/5	-6/5	3/5	-13/5
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-3/5	6/5	7/5	-27/5
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	0	-1	-1	-1

The solution $z^{*(2)} = (8/5, 1/5, 0)$ is not an integer, the branching process generates two nodes $node_1$, $node_2$ corresponding to the following constraints:
$node_1 : x_1 \geq \lceil \frac{8}{5} \rceil$.

$node_2 : x_1 \leq \lfloor \frac{8}{5} \rfloor$.
$node_1$: The constraint $x_1 \geq \lceil \frac{8}{5} \rceil$ is added to the $\mathtt{P}_2$ constraints, making the augmented problem unfeasible, then the node is fathomed.
$node_2$: The constraint $x_1 \leq \lfloor \frac{8}{5} \rfloor$ is added to the $\mathtt{P}_2$ constraints, we get $\mathtt{P}_3$, solve the problem, the integer optimal solution $z^{*(3)} = (1, 1/2, 0)$ is given in Table 4.

Table 4 – Simplex table for node 3.

B_3	x_6	x_9	x_{10}	Rhs
x_2	0	-1/2	-1/2	1/2
x_5	-1	-1/2	1/2	5/2
x_3	1	1	0	0
x_7	-2	-1/2	3/2	9/2
x_1	0	0	1	1
x_8	0	-1/2	1/2	1/2
x_4	-2	-3/2	-5/2	3/2
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-1	-2	-2	-4
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	3/2	0	3/2	-7/2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-3/2	0	1/2	-9/2
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	0	-1	-1	-1

The solution $z^{*(3)} = (1, 1/2, 0)$ is not an integer, the branching process generates two nodes $node_1$, $node_2$ corresponding to the following constraints:
$node_1 : x_2 \geq \lceil \frac{1}{2} \rceil$.

$node_2 : x_2 \leq \lfloor \frac{1}{2} \rfloor$.
$node_1$: The constraint $x_2 \geq \lceil \frac{1}{2} \rceil$ is added to $\mathtt{P}_3$ constraints, we get $\mathtt{P}_4$, solve the problem, the integer optimal solution $z^{*(4)} = (0, 1, 0)$ is given in Table 5.

Table 5 – Simplex table for node 4.

B_4	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

$z^{*(4)} = (0,1,0)$, the criterion vector is $(3,5,3,1)$, is not dominated, therefore we update $D_E = \{(0,0,2),(1,0,1),(0,1,0)\}$, $\bar{\mathscr{D}}_E = \emptyset$. $H_4 = \{9,11\}$, the efficient cut $x_9 + x_{11} \geq 1$ is added to $\mathtt{P}_4$ constraints, making the augmented problem unfeasible, then the node is fathomed.

$node_2$: The constraint $x_2 \leq \lfloor \frac{1}{2} \rfloor$ is added to the $\mathtt{P}_4$ constraints, constraints, which makes the augmented problem unfeasible. Consequently, the node is fathomed.

The efficient solution $z^{*(1)} \in D_E$ which maximize the value of the objective function of the problem $\mathtt{ULILMP}_0$ is $(1,0,1)$. Therefore, $z^{*(1)}$ is the optimal solution of the upper level $\mathtt{ULILMP}_0$ and $F_{opt} = 96$. $D_E^0 = D_E$, $\bar{D}_E^0 = \bar{D}_E$. goto step 3.

Step 3: Put $x_1 = x_1^{*(1)} = 1$ and solve the lower level problem $\mathtt{LLILMP}$ by the same technique cited in above Algorithm 2.

$$(\mathtt{LLILMP}) \begin{cases} \max f(1,y_1,y_2) = (y_1+1)(y_1-y_2+4) \\ s.t \\ s.t \\ y_1 + 2y_2 \quad \leq 2 \\ \qquad y_1 + y_2 \qquad \leq 3 \\ \quad 2y_1 + y_2 \quad \leq 1 \\ \qquad 3y_1 + 2y_2 \qquad \leq 6 \\ y_1 \geq 0, y_2 \geq 0, integers \end{cases} \tag{33}$$

The optimal solution is $(\tilde{x_2},\tilde{x_3}) = (0,0) \neq (0,1) = (x_2^1, x_3^1)$, then $z^{*(1)} = (1,0,1)$ is not optimal for the Integer Linear Multiplicative Bilevel Problem $\mathtt{ILMBP}$, goto step 4.

Step 4: Update the set D_E^0 and the set $\bar{D}_E^0$. $D_E^1 = D_E^0 \setminus \{z^{*(1)}\}$ and $\bar{D}_E^1 = \bar{D}_E^0 \cup \{z^{*(1)}\}$.

Step 5: $N_0 = \{2,6,8\}$. We reduce the feasible region of $\mathtt{ULILMP}_0$ using the cut $x_2 + x_6 + x_8 \geq 1$, we obtain the new problem $\mathtt{ULILMP}_1$ and solve it.

Step 2: Use the above Algorithm 2 to solve the problem $\mathtt{ULILMP}_1$. To do, add the cut $x_2 + x_6 + x_8 \geq 1$ to the program $\mathtt{P}_1$, corresponding to the solution $z^{*(1)} = (1,0,1)$ and use the dual simplex method to solve the new problem $\mathtt{P}_5$. The optimal solution $z^{*(5)} = (0,0,1)$ is given by Table 6:

Table 6 – Simplex table for node 5.

B_5	x_2	x_9	x_8	*Rhs*
x_4	-1	-3	4	3
x_5	0	0	-1	2
x_3	1	0	1	1
x_7	1	0	-2	4
x_1	0	1	-2	0
x_6	1	-1	1	1
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-2	0	-1	-5
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	3	0	0	-2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	1	4	-2	-2
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	-1	-1	0	-2

The criterion vector, which is dominated is $(5,2,2,2)$. Therefore, $D_E = \{(0,0,2),(0,1,0)\}$, $\bar{D}_E = \{(1,0,1),(0,0,1)\}$. $H_5 = \{2,8\}$, use the efficient cut $x_2 + x_8 \geq 1$, we determine a new program $\mathtt{P}_6$. The optimal solution $z^{*(6)} = (11/7, 0, 1/7)$ is indicated by Table 7:

Table 7 – Simplex table for node 6.

B_6	x_2	x_{10}	x_4	*Rhs*
x_9	5/7	-4/7	-1/7	1/7
x_5	2/7	-3/7	1/7	20/7
x_3	5/7	3/7	-1/7	1/7
x_7	11/7	-6/7	2/7	40/7
x_1	-1/7	-2/7	3/7	11/7
x_6	10/7	-1/7	-2/7	2/7
x_8	2/7	-3/7	1/7	6/7
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-9/7	-1/7	-4/7	-34/7
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	3	0	0	-2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	9/7	-6/7	4/7	-36/7
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	-5/7	1/7	-3/7	-8/7

As this optimal solution is not integer, the branching process generates two nodes $node_7$, $node_8$ corresponding to the following constraints:

$node_7 : x_1 \geq \lceil \frac{11}{7} \rceil$

$node_8 : x_1 \leq \lfloor \frac{11}{7} \rfloor$

$node_7$: The constraint $x_1 \geq \lceil \frac{11}{7} \rceil$ is added to Table 7, making the augmented problem unfeasible, then the node is fathomed.

node$_8$: The constraint $x_1 \leq \lfloor \frac{11}{7} \rfloor$ is added to Table 7 to yield Table 8 with optimal solution $z^{*(7)}$ = $(1, 0, \frac{1}{3})$

Table 8 – Simplex table for node 7.

B_7	x_2	x_{11}	x_{10}	Rhs
x_9	2/3	-1/3	-2/3	1/3
x_5	1/3	1/3	-1/3	8/3
x_3	2/3	-1/3	1/3	1/3
x_7	5/3	2/3	-2/3	16/3
x_1	0	1	0	1
x_6	4/3	-2/3	-1/3	2/3
x_8	1/3	1/3	-1/3	2/3
x_4	-1/3	-7/3	-2/3	4/3
$c^1_{1j} + c^1_{2j} - z^{1(1)}_j$	-4/3	-2/3	-1/3	-14/3
$d^1_{1j} + d^1_{2j} - z^{1(2)}_j$	3	0	0	-2
$c^2_{1j} + c^2_{2j} - z^{2(1)}_j$	1	0	-2	-4
$d^2_{1j} + d^2_{2j} - z^{2(2)}_j$	-2/3	-1/3	1/3	-4/3

As this optimal solution is not integer, the branching process generates two nodes N_9, N_{10} corresponding to the following constraints:

node$_9$: $x_3 \leq \lfloor \frac{1}{3} \rfloor$
node$_{10}$: $x_3 \geq \lceil \frac{1}{3} \rceil$
node$_9$: The constraint $x_3 \geq \lceil \frac{1}{3} \rceil$ is added to Table 8, making the augmented problem unfeasible, then the node is fathomed.
node$_{10}$: The constraint $x_3 \leq \lfloor \frac{1}{3} \rfloor$ is added to Table 8 to obtain Table 9 with optimal solution $z^{*(8)}$ = $(1, \frac{1}{2}, 0)$

Table 9 – Simplex table for node 8.

B_8	x_{12}	x_{10}	x_{11}	*Rhs*
x_9	1	-1	0	0
x_5	1/2	-1/2	1/2	5/2
x_3	1	0	0	0
x_7	5/2	-3/2	3/2	9/2
x_1	0	0	1	1
x_6	2	-1	0	0
x_8	1/2	-1/2	1/2	1/2
x_4	-1/2	-1/2	-5/2	3/2
x_2	-3/2	1/2	-1/2	1/2
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	0	-1	-2	-4
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	-3/2	3/2	9/2	-7/2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-1/2	-3/2	3/2	-9/2
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	0	0	-1	-1

As this optimal solution is not integer, the branching process generates two nodes $node_{11}$, $node_{12}$ corresponding to the following constraints:

$node_{11} : x_2 \leq \lfloor \frac{1}{2} \rfloor$

$node_{12} : x_2 \geq \lceil \frac{1}{2} \rceil$

$node_{11}$: *The constraint* $x_2 \leq \lceil \frac{1}{2} \rceil$ is added to Table 9 to yield Table 10 with optimal solution $z^{*(9)} = (1,0,0)$, the criterion vector is $(4,2,4,1)$, which is not dominated, therefore, we update $D_E = \{(0,0,2),(1,0,0),(0,1,0)\}$, $\bar{D}_E = \{(1,0,1),(0,0,1)\}$.

$H_9 = \emptyset$, then the node is fathomed.

$node_{12}$: *The constraint* $x_2 \geq \lfloor \frac{1}{2} \rfloor$ is added to Table 9 to yield Table 11 with optimal solution $z^{*(10)} = (0,1,0)$, the criterion vector is $(3,5,3,1)$, is not dominated, therefore, $D_E = \{(0,0,2),(1,0,0),(0,1,0)\}$, $\bar{D}_E = \{(1,0,1),(0,0,1)\}$.

$H_{10} = \{6,10,11\}$. The efficient cut $x_6 + x_{10} + x_{11} \geq 1$ is added to Table 11, making the augmented problem unfeasible, then the node is fathomed.

Table 10 – Simplex table for node 9.

B_9	x_{13}	x_{11}	x_{12}	*Rhs*
x_9	-2	-1	-2	1
x_5	-1	0	-1	3
x_3	0	0	1	0
x_7	-3	0	-2	6
x_1	0	1	0	1
x_6	-2	-1	-1	1
x_8	-1	0	-1	1
x_4	-1	-3	-2	2
x_2	1	0	0	0
x_{10}	-2	-1	-3	1
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-1	-2	0	-4
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	0	0	-3	-2
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-2	0	-1	-4
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	0	-1	0	-1

Table 11 – Simplex table for node 10.

B_{10}	x_{10}	x_6	x_{13}	*Rhs*
x_9	-1/2	-1/2	0	0
x_5	-1/2	1/2	1	2
x_3	1/2	-1/2	0	0
x_7	-1	1	3	3
x_1	-1/2	3/2	2	0
x_{11}	1/2	-3/2	-2	1
x_8	-1/2	1/2	1	0
x_4	1/2	-7/2	-5	4
x_2	0	0	-1	1
x_{12}	-1/2	1/2	0	0
$c^1_{1j}+c^1_{2j}-z^{1(1)}_j$	-1/2	-1/2	-2	-3
$d^1_{1j}+d^1_{2j}-z^{1(2)}_j$	0	0	3	-5
$c^2_{1j}+c^2_{2j}-z^{2(1)}_j$	-3	1	-3	-3
$d^2_{1j}+d^2_{2j}-z^{2(2)}_j$	1/2	-1/2	0	-1

The efficient solution $z^{*(0)} \in D_E$ which maximize the value of the objective function of the problem $\texttt{ULILMP}_1$ is $(0,0,2)$. Therefore, $z^{*(0)}$ is the optimal solution of the upper level $\texttt{ULILMP}_1$ and $F_{opt} = 84$. Goto step 3.

Step 3: Put $x_1 = x_1^{*(0)} = 0$ and solve the lower level problem LLILMP by the same technique cited in above Algorithm 2.

$$(\texttt{LLILMP}) \begin{cases} \max f(0,y_1,y_2) = (y_1+1)(y_1-y_2+3) \\ s.t \\ s.t \\ y_1+2y_2 \quad \leq 5 \\ \quad y_1+y_2 \quad \leq 3 \\ 2y_1+y_2 \quad \leq 2 \\ \quad 3y_1+2y_2 \quad \leq 6 \\ y_1 \geq 0, y_2 \geq 0, integers \end{cases} \tag{34}$$

The optimal solution is $(\tilde{x_2}, \tilde{x_3}) = (1,0) \neq (0,2) = (x_2^1, x_3^1)$, then
$z^{*(0)} = (0,0,2)$ is not optimal for the Integer Linear Multiplicative Bilevel Problem ILMBP, goto step 4.

Step 4: Update the set D_E^1 and the set $\bar{D}_E^1$. $D_E^2 = D_E^1 \backslash \{z^{*(0)}\}$ and $\bar{D}_E^2 = \bar{D}_E^1 \cup z^{*(0)}$.

There is a solution $(0,0,1) \in \bar{\mathscr{D}}_E^2$ such that the criterion vector $(5,2,2,2)$ is not dominated, then $\mathscr{D}_E^2 = \mathscr{D}_E^2 \cup \{(0,0,1)\}$ and $\bar{\mathscr{D}}_E^2 = \bar{\mathscr{D}}_E^2 \backslash \{(0,0,1)\}$, goto step 5.

Step 5: For $N_1 = \{1,2,6\}$ we reduce the feasible region of $\texttt{ULILMP}_1$ with the use of the cut $x_1+x_2+x_6 \geq 1$. We obtain the new problem $\texttt{ULILMP}_2$ and solve it.

Step 2: Use the above Algorithm 2 to solve the problem $\texttt{ULILMP}_2$. To do, add the cut $x_1+x_2+x_6 \geq 1$ to the program $\texttt{P}_0$, corresponding to the solution $z^{*(0)} = (0,0,2)$ and use the dual simplex method to solve the new problem $\texttt{P}_{11}$. The integer optimal solution $z^{*(11)}$ =$(1,0,1)$ is given in Table 12.

Table 12 – Simplex table for node 11.

B_{11}	x_2	x_6	x_8	Rhs
x_4	-4	-3	1	0
x_5	0	0	-1	2
x_3	1	0	1	1
x_7	1	0	-2	4
x_1	1	1	-1	1
$c_{1j}^1+c_{2j}^1-z_j^{1(1)}$	-3	-1	-1	-6
$d_{1j}^1+d_{2j}^1-z_j^{1(2)}$	3	0	0	-2
$c_{1j}^2+c_{2j}^2-z_j^{2(1)}$	-1	-2	2	-4
$d_{1j}^2+d_{2j}^2-z_j^{2(2)}$	-1	0	-1	-2

$F_{opt} < F(z^{*(11)})$, then the node is fathomed.

The efficient solution $z^{*(4)} \in D_E = \{(1,0,0),(0,1,0),(0,0,1)\}$ which maximize the value of the objective function of the problem $\mathtt{ULILMP}_2$ is $(0,1,0)$. Therefore, $z^{*(4)}$ is the optimal solution of the upper level $\mathtt{ULILMP}_2$ and $F_{opt} = 45$. Goto step 3.

Step 3: Put $x_1 = x_1^{*(0)} = 0$ and solve the lower level problem `LLILMP` by the same technique cited in above Algorithm 2.

$$(\mathtt{LLILMP}) \begin{cases} \max f(0,y_1,y_2) = (y_1+1)(y_1-y_2+3) \\ s.t \\ s.t \\ y_1 + 2y_2 \quad \leq 5 \\ \qquad y_1 + y_2 \qquad \leq 3 \\ \quad 2y_1 + y_2 \quad \leq 2 \\ \qquad 3y_1 + 2y_2 \qquad \leq 6 \\ y_1 \geq 0, y_2 \geq 0, integers \end{cases} \tag{35}$$

The optimal solution is $(\tilde{x_2},\tilde{x_3}) = (1,0) = (1,0) = (x_2^4,x_3^4)$, then
$z^{*(4)} = (0,1,0)$ **is the integer optimal solution of the Integer Linear Multiplicative Bilevel Problem `ILMBP`,** $F_{opt} = 45$ **and** $f_{opt} = 10$**.**

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

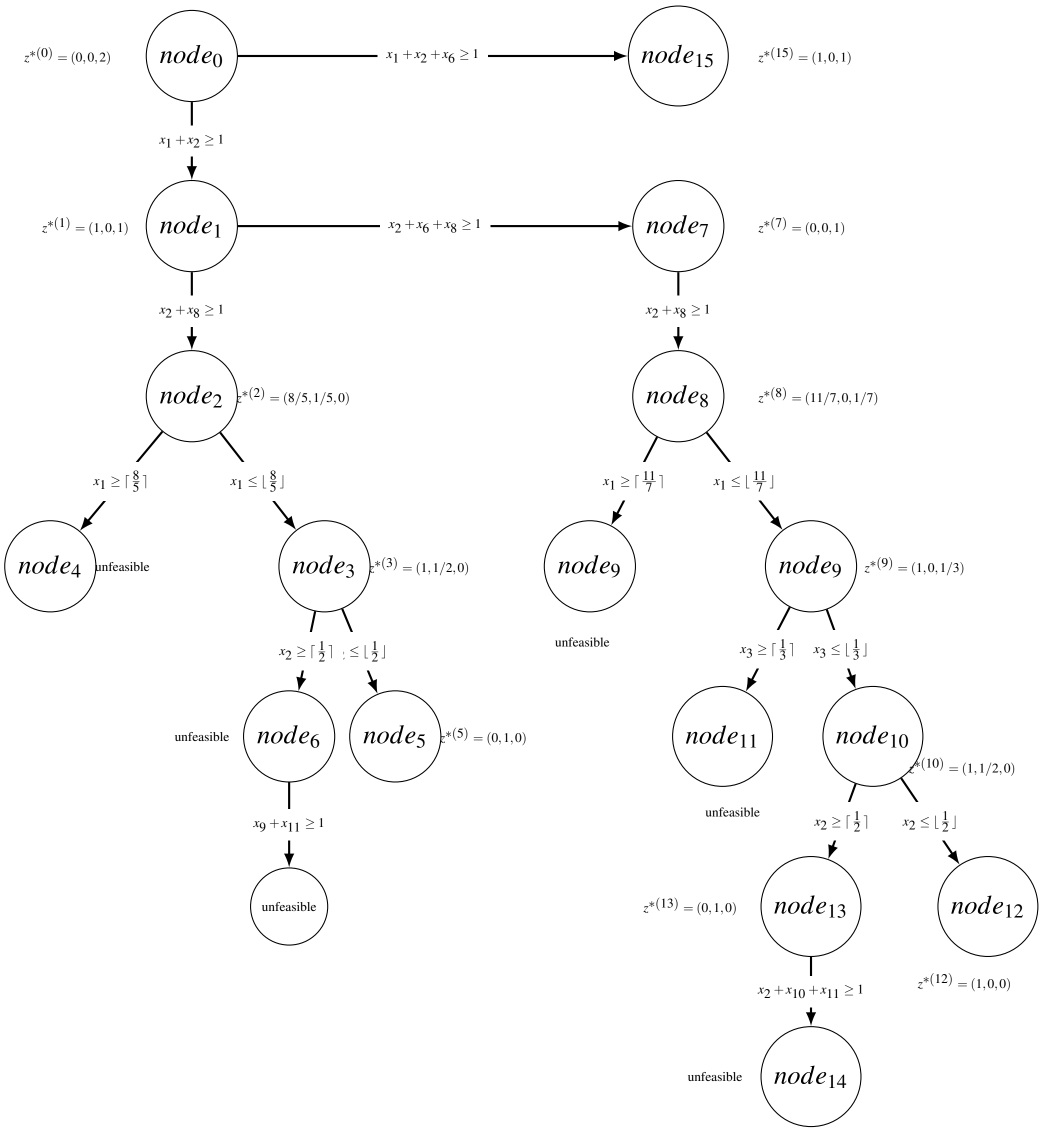

Figure 1 – Tree representing states of nodes during Branch and Cut algorithm for solving `ILMBP` problem.

6 NUMERICAL EXPERIMENTS

The method was implemented in the Visual Studio 2015 environment and tested on randomly generated (`ILMBP`) problems. The data is randomly generated by a discrete uniform distribution in the interval [1,30] for constraints coefficients, [50,100] for right hand side coefficient bj. The vectors $c_1^i, p_1^j, c_2^i, p_2^j$ and the scalars α^i, β^j are generated in [1,10] to ensure the necessary condition of positivity for $(c_1^i x + c_2^i y + \alpha^i)$ and $(p_1^j x + p_2^j y + \beta^j)$ for $i \in \{1,...,k\}$, $j \in \{1,...,l\}$. The Integer Linear Multiplicative Bilevel Programs are solved using the library `IBM CPLEX`

`12.8 for C++` programs. For each instance (n_1,n_2,m,k,l) (number of upper variables, number of lower variables, number of constraints, number of factors to the upper problem, number of factors to the lower problem), a series of 10 problems were solved. Computational experiments were carried out on a `2.4 GHZ ACER` computer, Intel(R) Core (TM) `i5` processor and `4 GB` memory. The obtained performance of the Algorithm 3 is summarized in Table 13 where mean, maximum number of CPU time (in seconds) are reported. As can be observed, the proposed Algorithm 3 is quick to execute (cpu(second)) for small and medium dimensions.

Table 13 – Computational results.

k	l	n_1	m	n_2	CPU(s)	
					Mean	Max
5	5	5	10	5	1.65	2.82
		10	10	10	13.72	18.7
		10	20	10	8.13	20.21
		20	20	10	38.30	56.23
		20	30	10	21.28	28.84
		20	30	20	47.06	72.66
		20	40	20	55.16	89.02
		20	40	30	128.91	220.98
		20	50	30	131.98	241.93
		30	50	30	249.84	279.76
		30	60	30	255.7	297.81
		40	60	30	559.91	738.83
		40	70	30	580.31	662.88
		50	70	30	1113.62	1228.86
		40	80	40	1217.02	1506.73
		40	80	50	2304.6	2526.94
		40	90	50	2416.11	2575.7
		40	90	60	4220.07	4419.8
		40	100	60	4368.74	4782.02
10	10	5	10	5	1.83	3.34
		10	10	10	13.1	25.37
		10	20	10	10.73	20.56
		20	20	10	35.88	62.69
		20	30	10	20.41	25.15
		20	30	20	81.75	153.84
		20	40	20	74.71	188.1
		20	40	30	190.87	297.93
		20	50	30	142.79	188.83
		30	50	30	318.63	398.29
		30	60	30	365.57	666.59
		40	60	30	705.42	838.83
		40	70	30	780.43	1139.49
		50	70	30	1375.11	1609.09
		40	80	40	1473.51	1694.83
		40	80	50	2917.5	3345.03
		40	90	50	2826.23	3234.92
		40	90	60	4833.1	5158.01
		40	100	60	5331.13	5656.49

7 CONCLUSION

In this study, we have presented a novel algorithm for solving the Integer Linear Multiplicative Bilevel Problem ILMBP. The algorithm is based on the branch and cut method. If the obtained integer optimal solution is not optimal for the main problem, a cut (27) is introduced that truncate the integer optimal solution of the upper problem (24), which allows us to select a new integer solution. The suggested algorithm offers the integer optimal solution, if one exists, and is an exact method.

The integer optimal solution of the main problem is reached in a finite number of iterations. We have provided a didactic example to explain the main steps of the algorithm. Moreover, to evaluate the algorithm, we conducted a numerical study where we calculated the computation time for random instances of different sizes and the results are acceptable.

Combinatorial linear multiplicative bilevel problems are a specific case of this mathematical program, hence our novel approach can be used to solve them. However, the technique can be improved by employing bounds in the branching process to determine the upper level's optimal solution more rapidly. This paper should spur researchers to create more effective methods for solving this issue. Future works might take into account the similar issue with nonlinear factors.

Acknowledgements

The authors are grateful to Dr Yacine Chaiblaine and anonymous reviewers whose comments allowed us to improve the manuscript significantly. The authors' work was supported by the General Direction for Scientific Research and Development Grant ID: C0656104.

References

ABBAS M & MOULAÏ M. 2002. Integer linear fractional programming with multiple objective. *Journal of the Italian Operations Research Society*, **32**(103-104): 15–38.

ANEJA Y, AGGARWAL VA & NAIR K. 1984. On a class of quadratic programs. *European Journal of Operational Research*, **18**(1): 62–70.

ARORA R & GUPTA K. 2018. Branch and bound algorithm for discrete multi-level linear fractional programming problem. *Operations Research and Decisions*, **28**.

ARORA R & GUPTA K. 2021. Fuzzy programming for multi-choice bilevel transportation problem. *Operations Research and Decisions*, **31**(3).

BECTOR C. 1972. Indefinite quadratic fractional functional programming. *Metrika*, **18**: 21–30.

CANDLER W & TOWNSLEY R. 1982. A linear two-level programming problem. *Computers & Operations Research*, **9**(1): 59–76.

CHAKRABARTY D, DEVANUR N & VAZIRANI VV. 2006. New results on rationality and strongly polynomial time solvability in eisenberg-gale markets. In: *Internet and Network Eco-*

nomics: Second International Workshop, WINE 2006, Patras, Greece, December 15-17, 2006. Proceedings 2. pp. 239–250. Springer.

CHERGUI MEA & MOULAÏ M. 2008. An exact method for a discrete multiobjective linear fractional optimization. *Journal of Applied Mathematics & Decision Sciences*, **2008**(1).

DEMPE S. 2002. *Foundations of bilevel programming*. Springer Science & Business Media.

DEMPE S & DUTTA J. 2012. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Mathematical programming*, **131**: 37–48.

DEMPE S, MORDUKHOVICH BS & ZEMKOHO AB. 2014. Necessary optimality conditions in pessimistic bilevel programming. *Optimization*, **63**(4): 505–533.

DEWEZ S, LABBÉ M, MARCOTTE P & SAVARD G. 2008. New formulations and valid inequalities for a bilevel pricing problem. *Operations research letters*, **36**(2): 141–149.

FORTUNY-AMAT J & MCCARL B. 1981. A representation and economic interpretation of a two-level programming problem. *Journal of the operational Research Society*, **32**: 783–792.

GOLPÎRA H. 2017. Robust bi-level optimization for an opportunistic supply chain network design problem in an uncertain and risky environment. *Operations Research and Decisions*, **27**(1): 21–41.

GOTOH JY & KONNO H. 2001. Maximization of the ratio of two convex quadratic functions over a polytope. *Computational Optimization and Applications*, **20**: 43–60.

IRELAND N. 1971. MICROECONOMIC THEORY-HENDERSON, JM AND QUANDT, RE.

JAIN K & VAZIRANI VV. 2007. Eisenberg-gale markets: Algorithms and structural properties. In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. pp. 364–373.

KONNO H & INORI M. 1989. Bond portfolio optimization by bilinear fractional programming. *Journal of the Operations Research Society of Japan*, **32**(2): 143–158.

KONNO H & KUNO T. 1992. Linear multiplicative programming. *Mathematical Programming*, **56**: 51–64.

MAACHOU N & MOULAÏ M. 2015. Bilevel quadratic fractional/quadratic problem. In: *Modelling, Computation and Optimization in Information Systems and Management Sciences: Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences-MCO 2015-Part I*. pp. 381–389. Springer.

MAACHOU N & MOULAÏ M. 2022. Branch and cut method for solving integer indefinite quadratic bilevel programs. *Annals of Operations Research*, pp. 1–31.

MALING K, HELLER W & MUELLER S. 1982. On finding most optimal rectangular package plans. In: *19th Design automation conference*. pp. 663–670. IEEE.

MARCHI E. 2008. *When is the product of two concave functions concave?* University of Minnesota. Institute for Mathematics and Its Applications.

MOULAÏ M & DRICI W. 2018. An indefinite quadratic optimization over an integer efficient set. *Optimization*, **67**(8): 1143–1156.

NARANG R & ARORA S. 2009. Indefinite quadratic integer bilevel programming problem with bounded variables. *Opsearch*, **46**: 428–448.

NASH JR J. 1950. The bargaining problem. Econometrica: J Econ Soc 18 (2): 155–162.

NEVES JCS, LUCAS SF, NEVES GC & WEISSMANN J. 2023. Bi-level model: new approach to dynamic vehicle allocation in supply chain. *Pesquisa Operacional*, **43**: e266695.

QUYNH TD, TAO PD ET AL. 2012. A DC programming approach for a class of bilevel programming problems and its application in portfolio selection. *Numerical Algebra, Control and Optimization*, **2**(1): 167–185.

SIMAAN M & CRUZ JR JB. 1973. On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, **11**(5): 533–555.

SINHA S. 2003. Fuzzy programming approach to multi-level programming problems. *Fuzzy sets and systems*, **136**(2): 189–202.

SWARUP K. 1966a. Indefinite quadratic programming. *Cahiers du Centre d'Etudes de Recherche Operationnelle*, **8**: 217–222.

SWARUP K. 1966b. Quadratic programming. *CCERO (Belgium)*, **8**(2): 132–136.

THIRWANI D & ARORA S. 1998. An algorithm for quadratic bilevel programming problem. *International Journal of Management and System*, **14**(2): 89–98.

VAZIRANI VV. 2012a. The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *Journal of the ACM (JACM)*, **59**(2): 1–36.

VAZIRANI VV. 2012b. Rational convex programs and efficient algorithms for 2-player Nash and nonsymmetric bargaining games. *SIAM Journal on Discrete Mathematics*, **26**(3): 896–918.

VICENTE LN & CALAMAI PH. 1994. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, **5**(3): 291–306.

WEN UP & HSU ST. 1991. Linear bi-level programming problems—a review. *Journal of the operational research society*, **42**: 125–133.

WEN UP & YANG Y. 1990. Algorithms for solving the mixed integer two-level linear programming problem. *Computers & Operations Research*, **17**(2): 133–142.

WHITE DJ & ANANDALINGAM G. 1993. A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, **3**: 397–419.

How to cite

MAACHOU N & MOULAÏ M. 2024. Branch-and-cut method for solving the Integer Linear Multiplicative Bilevel Problem. *Pesquisa Operacional*, **44**: e278191. doi: 10.1590/0101-7438.2023.043.00278191.