J. Math. Computer Sci., 25 (2022), 232–250

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

A simplex-based branch-and-cut method for solving integer tri-level programming problems

Ashenafi Awraris[a], Berhanu Guta Wordofa[a], Semu Mitiku Kassa[b,*]

[a] Department of Mathematics, Addis Ababa University, P.O. Box 1176, Addis Ababa, Ethiopia.

[b] Department of Mathematics and Statistical Sciences, Botswana International University of Science & Technology (BIUST), Botswana.

Abstract

Optimization problems involving three decision makers at three hierarchical decision levels are referred to as tri-level decision making problems. In particular in the case where all objective functions and constraints are linear, and all involved variables are required to take integer values is known as integer tri-level programming problems (T-ILP). It has been known that the general T-ILP is an NP-hard problem. So far there are very few attempts in the literature that finds an exact global solution for integer programming hierarchical problems beyond two levels. This paper proposes a novel approach based on a branch-and-cut (B&C) framework for solving T-ILP. The convergence of the algorithm is proved. Numerical examples are used to demonstrate the performance of the proposed algorithm. Finally, the results of this study show that the proposed algorithm is promising and more efficient to solve T-ILP. Moreover, it has been indicated in the remark that the proposed algorithm can be modified and used to solve discrete multilevel programming problems of any number of levels.

Keywords: Tri-level programming, integer programming, branch-and-cut, valid inequalities.

2020 MSC: 90C10, 90C57, 91A65.

1. Introduction

Many real-world decision problems involve multiple and independent decision makers (DMs), whose interests are not necessarily aligned. Optimization problems involving multiple decision makers, whose decisions are made in a sequential (or hierarchical) order are usually termed as *hierarchical or multilevel programming problems*. In such problems, it is assumed that decisions made earlier in the sequence affect the options available later in the sequence and the reactions from the lower levels affect the objective values of the upper level.

Hierarchical problems can be classified based on the number of hierarchical levels in the decision process and the type of decision variables in those levels. In particular, problems with three decision makers at three hierarchical decision levels are referred to as tri-level decision making problems (also known as *tri-level programming problems*). Decision entities at the three hierarchical levels are respectively

*Corresponding author

Email addresses: aawraris@gmail.com (Ashenafi Awraris), berhanu.guta@aau.edu.et (Berhanu Guta Wordofa), kassas@biust.ac.bw (Semu Mitiku Kassa)

doi: 10.22436/jmcs.025.03.03

Received: 2021-03-25 *Revised:* 2021-04-16 *Accepted:* 2021-05-20

termed as the top-level leader, the middle-level follower, and the bottom-level follower. In a tri-level decision-making process, the decision entities make their individual decisions in sequential order from the top level to the middle level and then to the bottom level with the aim of optimizing their respective objectives. Tri-level programming problems can be used to model practical problems in several areas, such as in supply chain management [22, 28], network defense [1, 30], planning [18], logistics [11], and economics [15] to mention a few. Therefore, investigating the properties and solution approaches of tri-level programming models has got a special attention in the last few decades.

Regarding solution approaches, many algorithms have been proposed in the open literature for solving bi-level problems as compared to the tri-level ones and the existing majority of research on tri-level programming has centered on the linear continuous version of the problem [16]. The discrete case, on the other hand, has not received mach attention in the open literature with a rather very limited number of studies. Generally, solution approaches presented in the literature for tri-level problems have addressed a very restricted class of problems (such as twice continuously differentiable problems in [8, 13, 14], or mainly linear continuous problems in [3, 25, 26, 31]), with only a few attempts to solve problems with integer variables. Since the algorithms and methods proposed so far are sensitive to the behaviour of the problem and are computationally expensive in some cases, it is more appealing to search for alternative methods that give exact global solutions to such problems and their generalizations.

This paper focuses on solving a generic integer tri-level linear programs (T-ILP), i.e., a tri-level optimization problem where all objective functions and constraints are linear, and all involved variables are required to take integer values. In addressing such kind of optimization problems, Sakawa [21] proposed a genetic algorithm to solve a special class of tri-level integer linear problems in 1999 and later on in 2014 Sakawa and Matsui [20] used a tabu search algorithm based on interactive fuzzy programming approaches to propose a solution strategy for 0-1 type problems. However, these methods produce approximate and sub-optimal solutions to the problem and finding an exact global optimal solution is essential in many application problems. More recently, Avraamidou and Pistikopoulos [2] proposed a parametric global solution of tri-level mixed-integer linear optimization problems containing both integer and continuous variables at all three optimization levels. Though the multi-parametric method gives exact global optimal solution, there are cases where the procedure becomes computationally expensive. Therefore, it is rather evident that alternative general strategies for the exact (global) solution of pure-integer tri-level linear problems are still lacking.

In this paper, we present a relatively simple branch-and-cut algorithm that solves general pure-integer tri-level programming problems. The main contributions of this paper are twofold. First, it extends the algorithm developed by DeNegre and Ralphs [7] that solves integer bi-level programming problems to effectively solve general integer tri-level programming problems. Second, it gives a way of extending the same method to solve any similar but higher level problems using the modified procedures.

The organization of the paper is as follows. Following the introduction, we present in Section 2 basic definitions and some theoretical properties of tri-level integer linear programming problems (T-ILP) that we shall investigate in the remaining sections. Based on the facts stated in Section 2, a branch-and-cut algorithm is proposed to solve the T-ILP in Section 3. Then, in Section 4, numerical examples for integer linear tri-level programming problems are provided. Finally, conclusive remarks are given in Section 5.

2. Basic definitions and theoretical background

In this section we introduce some definitions and explain the theoretical background for tri-level integer linear programming problems (T-ILP).

Let $x \in \mathbb{Z}^{n_1}$, $y \in \mathbb{Z}^{n_2}$, and $z \in \mathbb{Z}^{n_3}$ represent the variables controlled by the upper-level DM (the leader), the middle-level DM, and the lower-level DM, respectively. A linear tri-level integer programming problem is given by:

$$\begin{aligned}
&\min_{x}\ F_1(x,y,z) = a_1x + b_1y + c_1z\\
&\quad \text{s.t. } A_1x + B_1y + C_1z \leqslant b_1' \quad \text{where } [y,z] \text{ solves}\\
&\qquad \min_{y}\ F_2(x,y,z) = a_2x + b_2y + c_2z\\
&\qquad \text{s.t. } A_2x + B_2y + C_2z \leqslant b_2' \quad \text{where } z \text{ solves}\\
&\qquad\quad \min_{z}\ F_3(x,y,z) = a_3x + b_3y + c_3z\\
&\qquad\quad \text{s.t. } A_3x + B_3y + C_3z \leqslant b_3'\\
&\qquad\quad (x,y,z) \in \mathbb{Z}_+^{n_1} \times \mathbb{Z}_+^{n_2} \times \mathbb{Z}_+^{n_3},
\end{aligned} \tag{2.1}$$

where $a_i \in \mathbb{R}^{n_1}$, $b_i \in \mathbb{R}^{n_2}$, $c_i \in \mathbb{R}^{n_3}$ are row vectors, $b_i' \in \mathbb{R}^{q_i}$, $A_i \in \mathbb{R}^{q_i \times n_1}$, $B_i \in \mathbb{R}^{q_i \times n_2}$, $C_i \in \mathbb{R}^{q_i \times n_3}$ for $i = 1,2,3$.

Assume that $n = n_1 + n_2 + n_3$. Corresponding to problem (2.1), the following basic definitions of terms are important for the theoretical analysis of tri-level linear programming problem as in [3, 27].

Definition 2.1 (Regions)**.**

a. The **constraint region of the T-ILP** is given by

$$\Omega^I := \{(x,y,z) \in \mathbb{Z}_+^n : A_ix + B_iy + C_iz \leqslant b_i', \text{ for } i = 1,2,3,\}$$

i.e., it is the set of points that satisfy all the upper-level, the middle-level and the lower-level constraints.

b. For any given $x \in \mathbb{Z}_+^{n_1}$ the **feasible region of the second level** is

$$S^I(x) := \left\{(y,z) \in \mathbb{Z}_+^{n_2} \times \mathbb{Z}_+^{n_3} : B_2y + C_2z \leqslant b_2' - A_2x,\ B_3y + C_3z \leqslant b_3' - A_3x\right\},$$

i.e., the second level feasible region is affected by the leader's choice of variable x, and the second level allowable choices must be from the elements of Ω^I.

c. The **feasible region of the third level:** For any given $(x,y) \in \mathbb{Z}_+^{n_1} \times \mathbb{Z}_+^{n_2}$ the feasible set of the third level is

$$S^I(x,y) := \{z \in \mathbb{Z}_+^{n_3} : C_3z \leqslant b_3' - A_3x - B_3y\}.$$

d. The **rational reaction set of the third level for any given** (x,y) is

$$M^I(x,y) := \left\{\bar{z} : \bar{z} \in \underset{z}{\operatorname{argmin}}\{F_3(x,y,z), z \in S^I(x,y)\}\right\},$$

i.e., $M^I(x,y)$ is the set of third level optimal solutions for a given values of (x,y).

e. **The rational reaction set of the second level for any given choice** x **of the leader** is

$$M^I(x) := \{(\bar{y},z) : \bar{y} \in \underset{y}{\operatorname{argmin}}\{F_2(x,y,z) : (y,z) \in S^I(x), z \in M^I(x,y)\}\},$$

i.e., $M^I(x)$ is the set of second level optimal solutions corresponding to the solution x of the leader.

g. The set

$$\mathfrak{R}^I = \{(x,y,z) : (x,y,z) \in \Omega^I, (y,z) \in M^I(x)\}$$

is called the **inducible region** of problem (2.1).

Using the inducible region as given in Definition 2.1, the tri-level optimization problem (2.1) can be equivalently reformulated as

$$\begin{aligned}
\min\ & F_1(x,y,z)\\
\text{s.t. }\ & (x,y,z) \in \mathfrak{R}^I.
\end{aligned} \tag{2.2}$$

Notation 2.2. In this paper, we use Ω, $S(x)$, $S(x,y)$ $M(x)$, $M(x,y)$, and $\mathfrak{R}$ to denote the regions for which $(x,y,z) \in \mathbb{R}^{n_1}_+ \times \mathbb{R}^{n_2}_+ \times \mathbb{R}^{n_3}_+$ by removing the superscript I from the definition of problem (2.1).

A solution (x^*,y^*,z^*) is called *tri-level feasible* to the optimization problem (2.1) if $(x^*,y^*,z^*) \in \mathfrak{R}^I$ or equivalently if it satisfies the following four conditions:

(C1) $(x^*,y^*,z^*) \in \Omega$;
(C2) $(x^*,y^*,z^*) \in \mathbb{Z}^{n_1}_+ \times \mathbb{Z}^{n_2}_+ \times \mathbb{Z}^{n_3}_+$;
(C3) $(y^*,z^*) \in M^I(x^*)$;
(C4) $z^* \in M^I(x^*,y^*))$.

Moreover, a point $(\bar{x},\bar{y},\bar{z})$ is said to be an *optimal solution* to problem (2.2) if

i. $(\bar{x},\bar{y},\bar{z})$ is tri-level feasible; and
ii. $F_1(\bar{x},\bar{y},\bar{z}) \leqslant F_1(x,y,z)$, for all $(x,y,z) \in \mathfrak{R}^I$.

In this case, $F_1(\bar{x},\bar{y},\bar{z})$ is called the optimal value for problem (2.2).

In the formulation of problem (2.2), there is an ambiguity when multiple middle-level and lower-level optimal solutions exist. In the optimistic (or strong) formulation case, each of the middle-level and lower-level decision makers are assumed to select the one leading to the best from their optimal solution set according to the interest of the upper-level decision maker. On the contrary, in the pessimistic (or weak) formulation case, the middle-level and lower-level decision makers are assumed to select the one leading to the worst from among the possible optimal values of the upper level decision maker's interest. In this paper, we consider the optimistic formulation to avoid any such ambiguity.

In order to ensure the problem is well-posed and has a solution, our proposed approach requires the following assumptions.

Assumption 2.3.

A1. The constraint set Ω^I is bounded and nonempty.
A2. The feasible set $\mathfrak{R}^I$ is nonempty.
A3. For decisions taken by the leader, each of the followers have some room to respond, i.e., $M^I(x) \neq \emptyset$ and $M^I(x,y) \neq \emptyset$.

3. Branch-and-cut method for tri-level integer linear programming problems

There are various solution approaches proposed for bilevel discrete optimization problems. Some of the recent works can be obtained in [6, 7, 10, 19, 23, 29], where some of these methods often resulting in approximate solutions while others give exact solutions. However, due to the introduction of the middle-level follower in the problem, solution approaches developed for bi-level programming are not necessarily applicable to tri-level optimization problems. In this section we present modified procedures that will lead us to the proposed solution algorithm for tri-level integer linear programming problems.

3.1. Computational challenges in tri-level integer linear programming

A general branch-and-cut method for single level problems is a very successful algorithmic procedure for solving a variety of integer programming problems. Such a method functions based on the key ideas such as *separation*, *relaxation*, *bounding*, *fathoming*, and *branching* and using cutting planes to tighten the linear programming relaxations.

Those commonly used algorithmic frameworks can also be applied for tri-level integer linear programming problems followed by a process of removing its optimal solution if it is tri-level infeasible. But, the usual bounding, fathoming, and branching procedures employed in traditional linear programming

(LP)-based branch-and-bound algorithms for solving T-ILP cannot be applied in a straightforward way. To see this consider the following example.

$$\begin{aligned} &\min_{x_1} F_1(x_1,x_2,x_3) = -x_1 - 4x_2 - 5x_3, \quad \text{where} \quad [x_2,x_3] \;\; \text{solves} \\ &\min_{x_2} \; F_2(x_1,x_2,x_3) = 2x_3, \quad \text{where} \quad x_3 \;\; \text{solves} \\ &\min_{x_3} F_3(x_1,x_2,x_3) = -x_3 \\ &\quad \text{s.t.} \quad 3x_1 + 6x_2 + 3x_3 \leqslant 24 \\ &\qquad\quad x_1 + 2x_2 + 3x_3 \leqslant 12 \\ &\qquad\quad 3x_1 + 2x_2 + x_3 \geqslant 6 \\ &\quad (x_1,x_2,x_3) \in \mathbb{Z}^1_+ \times \mathbb{Z}^1_+ \times \mathbb{Z}^1_+. \end{aligned} \tag{3.1}$$

Figure 1 shows the half-space intersection of all the constraints and the non-negativity constraints for problem (3.1). Moreover, the values of all the vertices (corner points) of the solid and the corresponding upper level objective function values are given in Table 1.

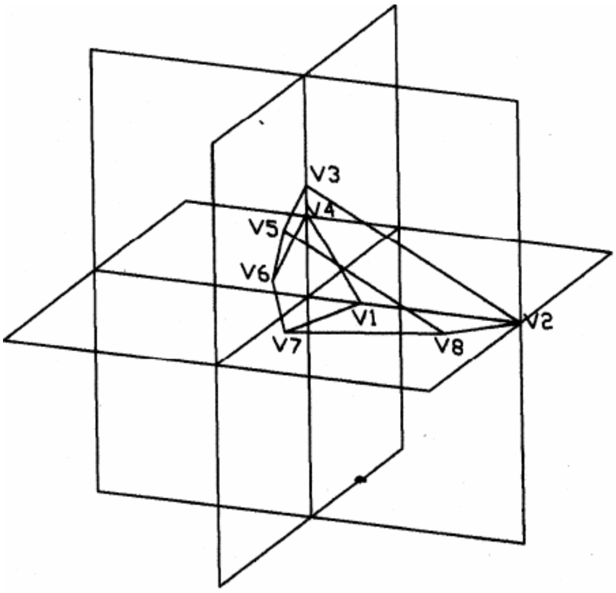

Figure 1: Solid obtained from the intersection of all the constraints of Eq. (3.1) as presented in [12].

Table 1: Feasible solution set and the respective objective function values.

S.No.	Vertices (corner points)	Coordinates	Objective Function value F_1
1	V1	(2,0,0)	-2
2	V2	(4,0,0)	- 8
3	V3	(0,4,0)	-16
4	V4	(0,3,0)	-12
5	V5	(0,3,2)	-22
6	V6	(0,1.5,3)	-21
7	V7	(7.5,0,0.5)	-10
8	V8	(6,0,2)	-16

First we make a reasonable effort to solve the original problem (Problem (3.1)) by considering a relaxation of it as:

$$\begin{aligned} \min \ & F_1(x,y,z) \\ \text{s.t.} \ & (x,y,z) \in \mathfrak{R}. \end{aligned}$$

It is well known [24] that optimal solution of the linear tri-level programming problem occurs at a vertex of the continuous induced region $\mathfrak{R}$ and also if (x_1,x_2,x_3) is an extreme-point of the induced region $\mathfrak{R}$, then it is an extreme-point of the constraint region Ω. However, this may not be true for pure-integer tri-level programming problems as the optimal solutions may occur in the interior of the tri-level feasible region. In this example, optimizing over the continuous region $\mathfrak{R}$, yields the integer solution $(x_1,x_2,x_3) = (6,0,2)$ (note that this point is in the inducible region) at the vertex V8, with the upper level objective value -16. But, the true solution of the problem with the extra requirements of integrality is $(x_1^*,x_2^*,x_3^*) = (1,1,3)$ in the interior of Ω with the upper-level objective value $F_1(x_1^*,x_2^*,x_3^*) = -20$. As a consequence, we have the following observations.

Observation 3.1.

i. A solution of the relaxed problem which is formed by removing integrality constraints does not provide a valid bound for the solution of the tri-level integer programming problem (3.1), i.e., $\mathfrak{R}^I \nsubseteq \mathfrak{R}$ – in general also to problem (2.1).

ii. Solutions to the relaxed problem that are in $\mathfrak{R}^I$ are not necessarily optimal solutions to Problem (3.1) (and hence also to problem (2.1)).

In the next subsections, we describe the procedure that overcomes these challenges to develop a generalized branch-and-cut algorithm for tri-level integer linear programming problems that follows the same basic paradigm used in bi-level and single level integer linear programming problems [7, 25].

3.2. *Relaxation of tri-level integer linear programming*

To solve the problem given in Eq. (2.1) it usually requires the solution of one or more easier sub-problems, where the integer restrictions and/or some of the other constraints are dropped. These easier problems are known as relaxations. To get these relaxations, first we convert a tri-level integer linear programming problem Eq. (2.1) as an equivalent single level problem using the optimal value reformulation (as in Outrata [17]), to get the following problem:

$$\begin{aligned} \min \ & F_1(x,y,z) && \text{(3.2a)} \\ \text{s.t.} \ & A_1x + B_1y + C_1z \leqslant b_1' && \text{(3.2b)} \\ & A_2x + B_2y + C_2z \leqslant b_2' && \text{(3.2c)} \\ & A_3x + B_3y + C_3z \leqslant b_3' && \text{(3.2d)} \\ & F_2(x,y,z) \leqslant \psi(x) && \text{(3.2e)} \\ & F_3(x,y,z) \leqslant \varphi(x,y) && \text{(3.2f)} \\ & (x,y,z) \in \mathbb{Z}_+^{n_1} \times \mathbb{Z}_+^{n_2} \times \mathbb{Z}_+^{n_3}, && \text{(3.2g)} \end{aligned}$$

where

$$\begin{aligned} \psi(x) &:= \min\{F_2(x,y,z) : (y,z) \in S^I(x), z \in M^I(x,y)\}, \\ \varphi(x,y) &:= \min\{F_3(x,y,z) : z \in S^I(x,y)\}. \end{aligned}$$

Proposition 3.2. *If (x^*,y^*,z^*) is an optimal solution of problem (3.2), then it is also an optimal solution of problem (2.1) and conversely. That is, the two problems (3.2) and (2.1) are equivalent.*

Proof. We first consider a feasible solution (x^*, y^*, z^*) to problem (3.2), and show that (x^*, y^*, z^*) is also feasible to problem (2.1). Constraints (3.2b)-(3.2d) and (3.2g) ensure that $(x^*, y^*, z^*) \in \Omega^I$. Note that constraints (3.2e) and (3.2f) enforce that $(y^*, z^*) \in M^I(x)$ and $z^* \in M^I(x, y)$. Hence tri-level feasibility condition is established. Thus (x^*, y^*, z^*) is feasible to the T-ILP (2.1). Because the objective function for both problems are the same, any feasible solution of (3.2) corresponds to that of (2.1). The converse statement is also not difficult to verify and hence is omitted here for brevity. □

An optimization problem obtained by dropping the optimality requirements (3.2e) and (3.2f) for the middle and follower variables of problem (3.2) is called the **high point relaxation (HPR)** [10], which is stated as

$$\begin{aligned} \min\ & F_1(x,y,z) \\ \text{s.t.}\ & A_1x + B_1y + C_1z \leqslant b_1' \\ & A_2x + B_2y + C_2z \leqslant b_2' \\ & A_3x + B_3y + C_3z \leqslant b_3' \\ & (x,y,z) \in \mathbb{Z}_+^{n_1} \times \mathbb{Z}_+^{n_2} \times \mathbb{Z}_+^{n_3} \end{aligned} \tag{\textbf{HPR}}$$

OR

$$\begin{aligned} \min\ & F_1(x,y,z) \\ \text{s.t.}\ & (x,y,z) \in \Omega^I. \end{aligned} \tag{\textbf{HPR}}$$

Clearly the inducible region $\mathfrak{R}^I$ is a subset of the feasible domain of the high point relaxation problem (**HPR**), i.e., $\mathfrak{R}^I \subseteq \Omega^I$.

The LP-relaxation of problem (**HPR**) which is obtained by dropping the integrality requirement on all or part of the variables is denoted by $\overline{\textbf{HPR}}$ and has the following form:

$$\begin{aligned} \min\ & F_1(x,y,z) \\ \text{s.t.}\ & (x,y,z) \in \Omega. \end{aligned} \tag{$\overline{\textbf{HPR}}$}$$

This problem has the following properties.

i. Since Ω is less constrained than Ω^I, the feasible domain of problem (**HPR**) is a subset of that of problem ($\overline{\textbf{HPR}}$), i.e., $\Omega^I \subseteq \Omega$.

ii. $\mathfrak{R}^I \subseteq \Omega^I \subseteq \Omega$.

Proposition 3.3. *If a point (x, y, z) is infeasible for the LP-relaxed problem ($\overline{\textbf{HPR}}$), then it is also infeasible for the original T-ILP problem.*

Proof. The proof obviously follows from the condition that $\mathfrak{R}^I \subseteq \Omega^I \subseteq \Omega$. □

3.3. *Branching*

Branching is a strategy for which unexplored nodes in the tree generate children by partitioning the solution space into smaller regions that can be solved recursively. An important question is how to branch, or how to split a subproblem into smaller subproblems. In this paper, we use the most common method called *single variable branching*. That is, at node t, pick an integer variable, say x^t, whose fractional part is the closest to 0.5 as the branching variable with the relaxed optimal solution component $\bar{x}^t$, and create branches by adding simple linear constraints $\bar{x}^t \leqslant \lfloor \bar{x}^t \rfloor$ and $\bar{x}^t \geqslant \lceil \bar{x}^t \rceil$.

And also, choosing a good branching variable is another crucial component of branch-and-bound algorithm. We denote by $I_c \in I$ the set of all candidate branching variables at a particular node in the branch-and-bound tree. A simple branching rule is to select the variable with the largest integer violation for branching, in other words, choose

$$\underset{i \in I_c}{\operatorname{argmax}} \{\min(\bar{x}_i^t - \lfloor \bar{x}_i^t \rfloor, \lceil \bar{x}_i^t \rceil - \bar{x}_i^t)\},$$

which is known as maximum fractional branching.

3.4. Bounding

The next fundamental process of the algorithm is formulation of the lower and upper bound of the problem at each given iteration. Bounding is determining a lower bound and upper bound to the optimal value of the given problem. Theses values are used to prune off regions of the search space that are suboptimal. Typically, lower bounds are obtained by solving LP-relaxation of the high point relaxation problem. Upper bounds are obtained by producing a tri-level feasible point (a point in the *inducible region* of the problem). An integer solution of **HPR** produces only a lower bound (but tighter than that of its relaxation) for the original problem. We denote the lower and upper bounds associated with node t by LB_t and UB_t, respectively. Whenever we cannot obtain a feasible solution to a given subproblem t, we set $UB_t = \infty$. Similarly, if the relaxation at node t is infeasible, we set $LU_t = -\infty$ and the node is pruned.

3.5. Pruning rules of tri-level integer linear programming

The pruning rules for T-ILP branch-and-cut are based on optimality and tri-level feasibility of subproblems. We let UB_t be an upper bound on the subproblem at node t (initialized as $UB_0 = \infty$).

1. **Infeasible nodes**. If the relaxation of the **HPR** for the subproblem at a node is infeasible, then any problem in the subtree rooted at this node is also infeasible. Thus, we can prune infeasible nodes.
2. **Integer feasible nodes**. If the solution, (x^t, y^t, z^t) of (**HPR**) satisfies Conditions (C1) to (C4) of the tri-level feasibility conditions (given on page 235), then we obtain a new incumbent solution if $F_1(x^t, y^t, z^t) < UB_t$, and we set $(x^*, y^*, z^*) = (x^t, y^t, z^t)$ and $UB_t = F_t(x^t, y^t, z^t)$. Otherwise, we prune the node because its solution is dominated by the upper bound.
3. **Upper bounds on T-ILP nodes**. If the optimal value of T-ILP $F_1(x^t, y^t, z^t)$ (or in fact a lower bound on the optimal value) is dominated by the upper bound (that is, if $F_1(x^t, y^t, z^t) \geqslant UB_t$), then we can prune this node because there cannot be any better integer solution in the subtree rooted at each level of the LP-relaxation problem.

3.6. Cutting procedure for tri-level integer linear programming

If a solution for (**HPR**) containing any fractional element, we apply the branching rules to remove such fractional infeasibility. However, if at node t a solution (x^t, y^t, z^t) of $\overline{\textbf{HPR}}_t$ satisfies integrality condition but not tri-level feasibility, we aim at deriving a cutting plane that will cut off the point (x^t, y^t, z^t) from the search region while not affecting all the points in $\mathfrak{R}^I$. To this end we introduce a valid inequality constraint into the problem $\overline{\textbf{HPR}}_t$ and re-solve it.

A *valid inequality* for an integer program (IP) is any constraint that does not eliminate any feasible integer solutions [5]. But, for a tri-level integer program, this is not the only case. A valid inequality for a tri-level integer program is any constraint that does eliminate a point which is integer feasible but not tri-level feasible. It is also called a cutting plane, or simply an intersection cut [10].

In tri-level integer linear programming, we are interested in finding such valid inequalities for $\mathfrak{R}^I$. One reason for this interest is derived from the relationship $\mathfrak{R}^I \subseteq \Omega^I$ and it allows us to separate fractional solutions to the LP resulting from the removal of the lower-level optimality and integrality restrictions. A second motivation is that we can separate points that are integer but not tri-level feasible, that is, we want intersection cuts that eliminate some members of Ω^I but leave all others that are in $\mathfrak{R}^I$.

Because the first two conditions (C1) and (C2) of the feasibility conditions (given on page 235) are enforced through the constraints of $\overline{\textbf{HPR}}_t$ and branching, the requirement for membership of the solution (x^t, y^t, z^t) in Ω^I is clear. Therefore, we must derive valid inequalities from the other two Conditions (C3) and (C4). But the process of defining such valid inequality depends on the type of the problem to be solved.

1. *Valid inequality*: For tri-level programming problems, in which all the problem data are integer (with possible exception of the objective function), the valid inequality as defined in [7] can be used to separate an integer solution (x^t, y^t, z^t) of the relaxed problem $\overline{\textbf{HPR}}_t$ from the set $\text{conv}(\mathfrak{R}^I)$. This cut can be derived by combining all inequalities that are binding at (x^t, y^t, z^t) as described in Proposition 3.4 below.

2. *Intersection cuts*: If the class of problem is not under the above category, one may consider using the idea of *intersection cuts* as described in [9]. This cut procedure is valid for problems that satisfy the conditions that $A_jx + B_jy + C_jz - b'_j$ are integer for $j = 2, 3$ at all solutions (x, y, z) of (**HPR**). The intersection cuts can be employed for removing the infeasible integer solution (x^t, y^t, z^t) from the feasible region, but it is not guaranteed when the lower level constraints are not integer at those points.

3. *Hypercube intersection cuts*: If the constraints of the lower level problems $A_jx + B_jy + C_jz - b'_j, j = 2, 3$ are not integer at the solution points, one may use the *hypercube intersection cut* as defined in Theorem 6 of [10]. The additional conditions in this cut are of the form $x^t_i - 1 \leqslant x^t \leqslant x^t_i + 1$ if Condition (C3) is violated, and additional constraints of the form $y^t_i - 1 \leqslant y^t \leqslant y^t_i + 1$ will be used if Condition (C4) is violated.

Proposition 3.4 (Generating Valid Inequalities)**.** *Let* $(\bar{x}, \bar{y}, \bar{z}) \in \Omega^I$ *be a basic feasible solution to* ($\overline{\textbf{HPR}}$)*, where all the problem data in* (**HPR**) *are integer. Let* J *be the set of indices of constraints that are binding at* $(\bar{x}, \bar{y}, \bar{z})$*, then*

$$\pi_1 x + \pi_2 y + \pi_3 z \leqslant \pi_0 - 1,$$

is valid for $\mathfrak{R}^I$*, where* $(\pi_1, \pi_2, \pi_3) = \sum_{i \in J}(a_i, b_i, c_i)$ *and* $\pi_0 = \sum_{i \in J} b'_i$*.*

Proof. The proof follows the same arguments as in [7].

The fact that $(\bar{x}, \bar{y}, \bar{z})$ is a basic feasible point implies that there exist $n = n_1 + n_2 + n_3$ linearly independent constraints in the description of Ω that are binding at $(\bar{x}, \bar{y}, \bar{z})$. Thus, the system $a_i^T x + b_i^T y + c_i^T z = b'_i$, $i \in J$ has a unique solution, namely $(\bar{x}, \bar{y}, \bar{z})$. This, in turn, implies that $(\bar{x}, \bar{y}, \bar{z})$ is the unique point of intersection between the hyperplane defined by the equation $\pi_1 x + \pi_2 y + \pi_3 z = \pi_0$ and the set Ω^I where $(\pi_1, \pi_2, \pi_3) = \sum_{i \in J}(a_i, b_i, c_i)$ and $\pi_0 = \sum_{i \in J} b'_i$. It follows that the inequality $\pi_1 x + \pi_2 y + \pi_3 z \leqslant \pi_0$ is valid for Ω. Because the face of Ω induced by this inequality does not contain any other members of Ω^I and there does not exist $(x, y, z) \in \mathbb{Z}^{n_1}_+ \times \mathbb{Z}^{n_2}_+ \times \mathbb{Z}^{n_3}_+$ such that $\pi_1 x + \pi_2 y + \pi_3 z \in (\pi_0 - 1, \pi_0)$, this implies that the inequality $\pi_1 x + \pi_2 y + \pi_3 z \geqslant \pi_0$ is valid for $\Omega^I \setminus (\bar{x}, \bar{y}, \bar{z})$. Then applying property (ii) of Observation 3.1 yields the required result. □

3.7. *Description of the proposed method*

Given a tri-level integer optimization problem (2.1), the goal of a branch-and-cut procedure is to iteratively build a search tree T of subproblems, or subsets of the search space to find an optimal solution

$$(x^*, y^*, z^*) \in \underset{(x,y,z) \in \mathfrak{R}^I}{\operatorname{argmin}} \; F_1(x, y, z).$$

As a first step in this procedure, one needs to solve the LP-relaxation of the high point problem:

$$\begin{aligned} \min \;& F_1(x, y, z) \\ \text{s.t.} \;& (x, y, z) \in \Omega \end{aligned} \tag{$\overline{\textbf{HPR}}$}$$

using any linear programming solver to get a solution $(\bar{x}_1, \bar{y}_1, \bar{z}_1)$.

If the solution to ($\overline{\textbf{HPR}}$) is infeasible (i.e., if condition (C1) is violated), then the T-ILP (2.1) is also infeasible and hence the process will be terminated according to the assertion in Proposition 3.3. Otherwise, if it is feasible but $(\bar{x}_1, \bar{y}_1, \bar{z}_1) \notin \mathbb{Z}^n_+$ (i.e., tri-level feasibility condition (C2) is violated), then apply ILP-**Branch** as indicated in Subsection 3.3 and solve the branched ($\overline{\textbf{HPR}}$) problems again in each branch. On the other hand, if the solution satisfies integer feasibility conditions (C1) & (C2), then we set the lower bound to be $LB_t = F_1(\bar{x}_1, \bar{y}_1, \bar{z}_1)$ and check whether it satisfies tri-level feasibility conditions (C3) and (C4).

To check tri-level feasibility condition (C3), fix $x := \bar{x}^1$ and solve the middle level problem

$$\min_{(y,z) \in \Omega \cap \{x : x = \bar{x}\}} F_2(x, y, z), \tag{\textbf{LLP}}$$

using any linear programming solver to get a solution $(\bar{y}_2, \bar{z}_2)$.

Let the solution to this (**LLP**) be $(\bar{y}_2, \bar{z}_2)$. Similarly, if this solution to (**LLP**) is infeasible, additional investigation is needed and the current process will be terminated. Otherwise, if it is feasible but $(\bar{y}_2, \bar{z}_2) \notin \mathbb{Z}_+^{n_2} \times \mathbb{Z}_+^{n_3}$, then apply ILP-Branch to separate at $(\bar{x}_1, \bar{y}_2, \bar{z}_2)$ and solve (**LLP**) again in each corresponding branch. On the other hand, if the solution satisfies integer feasibility, then we must check whether it satisfies tri-level feasibility conditions.

If the point $(\bar{x}_1, \bar{y}_2, \bar{z}_2)$ is feasible but $F_2(\bar{x}_1, \bar{y}_1, \bar{z}_1) \neq F_2(\bar{x}_1, \bar{y}_2, \bar{z}_2)$, then we generate a valid inequality (as outlined in Subsection 3.6) and incorporate it to ($\overline{\textbf{HPR}}$). With this modification, the problem is re-optimized to improve the bound for ($\overline{\textbf{HPR}}$). On the other hand, if $F_2(\bar{x}_1, \bar{y}_1, \bar{z}_1) = F_2(\bar{x}_1, \bar{y}_2, \bar{z}_2)$, then $(\bar{y}_2, \bar{z}_2)$ is optimal for (**LLP**) and we conclude that $(\bar{x}_1, \bar{y}_2, \bar{z}_2)$ is feasible for the second level problem and satisfies condition (C3).

If $(\bar{x}_1, \bar{y}_2, \bar{z}_2)$ satisfies conditions (C1), (C2), and (C3), then we must check whether it also satisfies condition (C4). This is done by fixing $(x, y) := (\bar{x}_1, \bar{y}_2)$ and solving the third level problem:

$$\min_{z \in \Omega \cap \{x : (x,y) = (\bar{x}_1, \bar{y}_2)\}} F_3(x, y, z). \tag{\textbf{LLLP}}$$

Assume that the solution to this (**LLLP**) be $\bar{z}_3$. Similar to the middle follower step, if the solution to (**LLLP**) is infeasible, the node will be pruned. In addition, if $\bar{z}_3$ is not an integer, one needs to apply the branching rule until we obtain integer $\bar{z}_3$. If $F_3(\bar{x}_1, \bar{y}_2, \bar{z}_2) = F_3(\bar{x}_1, \bar{y}_2, \bar{z}_3)$, then $\bar{z}_3$ is optimal for (**LLLP**) and we can conclude that $(\bar{x}_1, \bar{y}_2, \bar{z}_3)$ is tri-level feasible, and the optimal solution is obtained for the current node, and the node is pruned. Otherwise, we must again generate a valid inequality (or a cutting plane) separating $(\bar{x}_1, \bar{y}_2, \bar{z}_3)$ from $\text{conv}\mathfrak{R}^I$. Now suppose $F_3(\bar{x}_1, \bar{y}_2, \bar{z}_2) \neq F_3(\bar{x}_1, \bar{y}_2, \bar{z}_3)$. In this case, $(\bar{x}_1, \bar{y}_2, \bar{z}_3)$ does not satisfy condition (C4) and is therefore not feasible for the third level. We may still use $(\bar{x}_1, \bar{y}_2, \bar{z}_3)$ to update the upper bound of the original problem, but we would like to add an inequality to (**LLLP**) that is valid for Ω^I and is violated by $(\bar{x}_1, \bar{y}_2, \bar{z}_3)$ using the procedure in Subsection 3.6 and the problem is re-optimized to improve the bound for ($\overline{\textbf{HPR}}$). This process repeats until a final optimal solution is obtained.

The above procedure that solves integer tri-level programming problems is summarized in Algorithm 1.

Proposition 3.5. *The procedures described in Algorithm 1 terminate at a global optimal solution of problem* (2.1) *in a finite number of steps.*

Proof. First observe that every discrete collection of non-empty bounded sets of Ω_t^I is finite because Ω^I is countably compact. Then, as mentioned in ($\overline{\textbf{HPR}}$), each of the problems ($\textbf{LP}_t$), and ($\textbf{LLP}_t$) has an optimal solution at an extreme point of LP-relaxation of $(\Omega_t^i)^I \subseteq \Omega^I$ for $i = 0, 1, 2$ or has no solution. Then all the integer constrained variable vectors x, y and z have finite lower and upper bounds in ($\overline{\textbf{HPR}}$), ($\textbf{LP}_t$), and ($\textbf{LLP}_t$). Branching and generating valid inequality steps 2.2, 3.3, 3.3.1, 4,2 and 4.3.1 of Algorithm 1 on the tightness of the follower's constraints imply an implicit enumeration of all bases of $(\Omega_t^i)^I$, which occur finite number of times. Moreover, the steps do not cycle as they are done in a sequence unless simplification or branching takes place, which can happens again only a finite number of times.

In addition, the while loop that begins at Step 2 of Algorithm 1, completes after at most $|\Omega^I| + 1$ iterations. Then, by the finiteness of Ω^I, there must be two iterations i and k, $1 \leqslant i < k \leqslant |\Omega^I| + 1$, such that $UB_i = F_1(x^*, y^*, z^*) = LB_k$. Therefore, Algorithm 1 reaches at the condition $UB_k = LB_k$ in iteration k, and terminates with solution (x_k^*, y_k^*, z_k^*) after iteration k. Because the algorithm terminates in no more than $|\Omega^I| + 1$ iterations with a tri-level feasible solution whose objective value equals an upper bound on $F_1(x^*, y^*, z^*)$, Algorithm 1 terminates finitely with a global optimal solution. □

Remark 3.6. Even if the scope of this paper is to solve tri-level integer programming problems, the same algorithm can be easily extended to solve higher level (multilevel) programming problems with the same formulations and that satisfy the necessary assumptions in the article. This can be achieved by applying repeatedly and recursively Steps 3 & 4 of Algorithm 1 for any lower level problems, if there are some beyond the third level.

Algorithm 1: Pseudo code for the Branch-and-cut algorithm.

```
1. (Initialization) Initialize $UB = UB_0 = \infty$, $LB_0 = -\infty$, and set counter $t = 1$
2. while $UB > LB_t$ Do :
   Solve the top level $(\overline{\mathbf{HPR}}_t)$ to get a solution $(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t)$
       $\min \; F_1(x, y, z)$
       $\text{s.t.} \; (x, y, z) \in \Omega_t$          $(\overline{\mathbf{HPR}}_t)$
   2.1 If $(\overline{\mathbf{HPR}}_t)$ is infeasible or $F_1(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t) \geqslant UB$, then prune it due to infeasibility;
   2.2 else if $(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t) \notin \mathbb{Z}_+^n$, then apply (ILP) cutting plane techniques or Branch to separate $(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t)$
       from $\Omega_t^I$ and re-solve $(\overline{\mathbf{HPR}}_t)$ with the new branches.
   2.3 else if conditions (C1) and (C2) are satisfied, set $LB_t = F_1(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t)$ and go to Step 3.
3. Fix $\bar{x}_1^t \longleftarrow x$, set $\Omega_t^2 = \Omega_t \cap \{x : x = \bar{x}_1^t\}$ and solve the relaxed problem of $(\mathbf{LP}_t)$:
       $\min_{(y,z) \in (\Omega_t^2)^I} F_2(x, y, z)$          $(\mathbf{LP}_t)$
   to get $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t)$.
   3.1 If the relaxation of problem $(\mathbf{LP}_t)$ is infeasible, prune.
   3.2 else if $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t) \notin \mathbb{Z}_+^n$, then apply (ILP) cutting plane technique or Branching to separate $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t)$
       from $(\Omega_t^2)^I$ and re-solve the relaxation of problem $(\mathbf{LP}_t)$ in each of the two branches.
   3.3 else $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t) \in \mathbb{Z}_+^n$ and:
       3.3.1 If $F_2(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t) = F_2(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t)$ and condition (C3) are all satisfied, then go to Step 4.
       3.3.2 Otherwise, update $UB_t = F_1(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t)$ and construct a cutting plane (valid inequality) using the
             technique in Subsection 3.6 and go to Step 2.
4. Fix $(\bar{x}_1^t, \bar{y}_1^t) \longleftarrow (x, y)$, set $\Omega_t^3 = \Omega_t^2 \cap \{(x, y) : (x, y) = (\bar{x}_1^t, \bar{y}_1^t)\}$ and solve the relaxation of
       $\min_{z \in (\Omega_t^3)^I} F_3(x, y, z)$          $(\mathbf{LLP}_t)$
   to get $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t)$.
   4.1 If the relaxation of problem $(\mathbf{LLP}_t)$ is infeasible, prune.
   4.2 else if $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t) \notin \mathbb{Z}_+^n$, then apply (ILP) cutting plane techniques or Branch to separate $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t)$
       from $(\Omega_t^3)^I$ and solve $(\mathbf{LLP}_t)$ again.
   4.3 else $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t) \in \mathbb{Z}_+^n$ and:
       4.3.1 If $F_3(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t) = F_3(\bar{x}_1^t, \bar{y}_1^t, \bar{z}_1^t)$ and condition (C3) is satisfied, update $UB_t = F_1(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t)$ and set
             the incumbent solution to be $(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t) \longleftarrow (\bar{x}_1^t, \bar{y}_2^t, \bar{z}_2^t)$, update $UB = \min\{UB, UB_t\}$, and prune by
             optimality.
       4.3.2 Otherwise, update $UB_t = F_1(\bar{x}_1^t, \bar{y}_2^t, \bar{z}_3^t)$ and construct a polyhedron (valid inequality) using the
             technique in Subsection 3.6 and go to Step 2.
5. t=t+1;
6. end while
7. Return $(\bar{x}^t, \bar{y}^t, \bar{z}^t)$.
```

4. Numerical examples

To demonstrate the performance of the proposed branch-and-cut algorithm, the following 3 examples are formulated and solved using the procedures outlined in the previous section.

Example 4.1. Consider the following discrete tri-level programming problem:

$$\begin{aligned}
\min_{x_1,x_2,x_3}\quad & F_1 = -7x_1 + y_1 - 2x_2 \\
\text{s.t.} \min_{y_1,y_2}\quad & F_2 = -x_1 - 3y_1 - x_2 - 3y_2 + x_3 + 5z_1 + 2z_2 \\
\text{s.t.} \min_{z_1,z_2}\quad & F_3 = 2x_1 + x_2 + 2x_3 + y_1 - 4x_2 + 2z_1 + 10z_2, \\
\text{s.t.}\quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5, \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5, \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1, \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1, \\
& -10 \leqslant x_1, x_2 \leqslant 10, \\
& x_1, x_2, x_3 \in \mathbb{Z}^3, y_1, y_2, z_1, z_2 \in \{0,1\}^4.
\end{aligned} \tag{4.1}$$

This problem also appears in [2] but in mixed integer tri-level programming form. We solve here below the discrete version of this problem applying the procedures in Algorithm 1.

Iteration 1: Initialization: $UB_0 = \infty$, $LB_0 = -\infty$, and set counter $t = 1$.

Step 1: (Relaxed solution) Solve the top level high point problem

$$\begin{aligned}
\min\quad & F_1 = -7x_1 + y_1 - 2x_2 \\
\text{s.t.}\quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1
\end{aligned} \tag{4.2}$$

using any linear programming solver to get the solution

$$(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^1, \bar{y}_2^1, \bar{z}_1^1, \bar{z}_2^1) = (10, -8.2371, -13.9494, 0, 1, 0, 0),$$

and with the corresponding objective value $F_1 = -53.5258$.

Now, since $\bar{x}_2$ must take an integer value in (4.2), the optimal solution has to satisfy either the condition $x_2 \leqslant -9$ or $x_2 \geqslant -8$. It follows that the optimal solution of (4.2) will be the best among the optimal solutions of the subproblems (**LR1**) and (**LR2**) which are defined as follows:

$$\begin{aligned}
& \min F_1 = -7x_1 + y_1 - 2x_2 \\
& \text{s.t. } 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 \leqslant -9 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1,
\end{aligned} \tag{4.3}$$

$$\begin{aligned}
\min\ & F_1 = -7x_1 + y_1 - 2x_2 \\
\text{s.t. } & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 \geqslant -8 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1.
\end{aligned} \tag{4.4}$$

The above procedure is a branching operation on variable $\bar{x}_2$.

By solving sub-problem (**LR1**), we determine that there is no feasible solution and hence (4.3) contains no integer solution. Therefore, the branch is pruned by infeasibility. But, solving the sub-problem (4.4), we obtain the optimal solution of the sub-problem to be $(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^1, \bar{y}_2^1, \bar{z}_1^1, \bar{z}_2^1) = (10, -8, -14.3125, 0, 1, 0, 0)$ with objective value $F_1 = -54.0000$. Since this solution is also not an integer, we branch on $\bar{x}_3$ to obtain the subproblems (**LR3**) and (**LR4**) shown below:

$$\begin{aligned}
\min\quad & F_1 = -7x_1 + y_1 - 2x_2 \\
\text{s.t.}\quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 \geqslant -8, \ x_3 \leqslant -15 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1,
\end{aligned} \tag{4.5}$$

$$\begin{aligned}
\min\quad & F_1 = -7x_1 + y_1 - 2x_2 \\
\text{s.t.}\quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 \geqslant -8, \ x_3 \geqslant -14 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1.
\end{aligned} \tag{4.6}$$

Here, after solving the sub-problem (4.5), we determine that it has no feasible solution. That means, (4.5) contains no integer solution and then the subproblem is pruned by infeasibility. But, solving the sub-problem (4.6), we obtain the optimal solution : $(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^1, \bar{y}_2^1, \bar{z}_1^1, \bar{z}_2^1) = (10, -8, -14, 0, 1, 0, 0)$ with the corresponding objective value of $F_1 = -54.0000$. Since this is an integer solution, we consider it as possible solution for this branch.

The Branch-and-Bound tree, does not have any active node and therefore the incumbent solution is the best integer solution of the problem, which is $(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^1, \bar{y}_2^1, \bar{z}_1^1, \bar{z}_2^1) = (10, -8, -14, 0, 1, 0, 0)$ with objective value $F_1 = -54$, that is obtained at (4.6) branch. Observe that this solution satisfies conditions (C1) and (C2) and so we update $LB_1 = F_1 = -54$ and the procedure goes to **Step 2**.

Step 2: Check whether the last solution satisfies condition (C3). To do this:

- Fixing $(x_1, x_2, x_3) \longleftarrow (10, -8, -14)$ in the middle level problem we solve

$$\begin{aligned}
\min \quad & F_2 = -x_1 - 3y_1 - x_2 - 3y_2 + x_3 + 5z_1 + 2z_2 \\
\text{s.t.} \quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 \geqslant -8, \ x_3 \geqslant -14 \\
& x_1 = 10, x_2 = -8, x_3 = -14 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \quad 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1,
\end{aligned} \tag{4.7}$$

using any linear programming solver to get $(\bar{y}_1^2, \bar{y}_2^2, \bar{z}_1^2, \bar{z}_2^2) = (0, 1, 0, 0)$. Since, $F_2(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^2, \bar{y}_2^2, \bar{z}_1^2, \bar{z}_2^2) = -19 = F_2(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{y}_1^1, \bar{y}_2^1, \bar{z}_1^1, \bar{z}_2^1)$, then $(x_1, x_2, x_3) = (10, -8, -14)$ and $(y_1, y_2, z_1, z_2) = (0, 1, 0, 0)$ is also optimal for (4.7) and we can conclude that it is also feasible for the second level and satisfies condition (C3) and so update $UB_1 = F_1 = -54$ and go to the next step, **Step 3**.

Step 3: Check whether it satisfies condition (C4). To this end we have:

- Fix $(x_1, x_2, x_3, y_1, y_2) \longleftarrow (10, -8, -14, 0, 1)$ in the lower level problem and get the modified lower level problem:

$$\begin{aligned}
\min \quad & F_3 = 2x_1 + x_2 + 2x_3 + y_1 - 4x_2 + 2z_1 + 10z_2 \\
\text{s.t.} \quad & 6.4x_1 + 7.2x_2 + 2.5x_3 \leqslant 11.5 \\
& -8x_1 - 4.9x_2 - 3.2x_3 \leqslant 5 \\
& 3.3x_1 + 4.1x_2 + 0.002x_3 + 0.2y_1 + 0.8y_2 + 4z_1 + 4.5z_2 \leqslant 1 \\
& y_1 + y_2 + z_1 + z_2 \geqslant 1 \\
& x_2 <= -8, x_3 >= -14 \\
& x_1 = 10, x_2 = -8, x_3 = -14 \\
& y_1 = 0, y_2 = 1 \\
& -10 \leqslant x_1, x_2 \leqslant 10 \\
& x_1, x_2, x_3 \in \mathbb{R}^3, \ 0 \leqslant y_1, y_2, z_1, z_2 \leqslant 1.
\end{aligned} \tag{4.8}$$

After solving the problem (4.8) using any linear programming solver we get the solution $(\bar{z}_1^3, \bar{z}_2^3) = (0, 0)$. Since, $F_3(\bar{x}_2^2, \bar{x}_2^2, \bar{x}_3^3, \bar{y}_2^2, \bar{z}_1^3, \bar{z}_2^3) = 61 = F_3(\bar{x}_2^2, \bar{x}_2^2, \bar{x}_3^2, \bar{y}_2^2, \bar{z}_1^2, \bar{z}_2^2)$, the point $(10, -8, -14, 0, 1, 0, 0)$ satisfies condition (C4) and so we update $UB_1 = F_1 = -54$.

Since $UB_1 = LB_1$, the Algorithm terminates and the point

$$(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}_1, \bar{y}_2, \bar{z}_1, \bar{z}_2) = (10, -8, -14, 0, 1, 0, 0)$$

is the global optimal solution of the problem presented in Eq. (4.1).

As we can see from the solution procedures, the problem in Eq. (4.1) does not require the use of valid inequalities (or intersection cuts) to arrive at the global optimal solution. In the next example we will demonstrate the introduction and use of such cutting hyperplanes.

Example 4.2. Consider the following integer tri-level programming problem:

$$\begin{aligned}
\min_{x} \quad & F_1(x,y,z) = -x-4y-5z, \quad \text{where } [y,z] \text{ solves}\\
\min_{y} \quad & F_2(x,y,z) = -2y, \quad \text{where } z \text{ solves}\\
\min_{z} \quad & F_3(x,y,z) = z\\
\text{s.t.} \quad & 3x+6y+3z \leqslant 24\\
& x+2y+3z \leqslant 12\\
& 3x+2y+z \geqslant 6\\
& (x,y,z) \in \mathbb{Z}^1_+ \times \mathbb{Z}^1_+ \times \mathbb{Z}^1_+.
\end{aligned} \tag{4.9}$$

The procedures of the solution are presented in the following detailed steps.

Iteration 1: Initialization: set $UB_0 = \infty$, $LB_0 = -\infty$, and put the counter $t = 1$.

Step 1: Relaxed HPR: Solve the relaxed high point problem

$$\begin{aligned}
\min\ & F_1(x,y,z) = -x-4y-5z\\
\text{s.t. } & 3x+6y+3z \leqslant 24\\
& x+2y+3z \leqslant 12\\
& 3x+2y+z \geqslant 6\\
& (x,y,z) \in \mathbb{R}^1_+ \times \mathbb{R}^1_+ \times \mathbb{R}^1_+
\end{aligned} \tag{4.10}$$

using any linear programming solver (e.g., the simplex method) to get the solution $(\bar{x}_1, \bar{y}_1, \bar{z}_1) = (0,3,2)$, with the corresponding objective value $F_1 = -22$. Since this solution satisfies conditions (C1) and (C2), we update $LB_1 = F_1 = -22$ and go to the next step.

Step 2: Fix $x \longleftarrow 0$ in the middle level problem and solve

$$\begin{aligned}
\min\ & F_2(x,y,z) = -2y\\
\text{s.t. } & 3x+6y+3z \leqslant 24\\
& x+2y+3z \leqslant 12\\
& 3x+2y+z \geqslant 6\\
& x = 0\\
& (x,y,z) \in \mathbb{R}^3_+
\end{aligned}$$

using any linear programming solver, to get the solution $(\bar{x}_1, \bar{y}_2, \bar{z}_2) = (0,4,0)$ with the corresponding objective value of $F_2(\bar{x}_1, \bar{y}_2, \bar{z}_2) = -8$. Since $F_2(\bar{x}_1, \bar{y}_2, \bar{z}_2) = -8 < -6 = F_2(\bar{x}_1, \bar{y}_1, \bar{z}_1)$, which means $F_2(\bar{x}_1, \bar{y}_2, \bar{z}_2) \neq F_2(\bar{x}_1, \bar{y}_1, \bar{z}_1)$, the current solution is not optimal or does not satisfy condition (C3). So, we update $UB_t = F_1(\bar{x}^t_1, \bar{y}^t_2, \bar{z}^t_2) = -16$ and generate an inequality which is violated by $(0,3,2)$ and valid for the other feasible points of problem (4.10).

As all the data in problem (4.9) are integer, we can apply the valid inequality technique as described in Subsection 3.6. Therefore, since $3x+6y+3z \leqslant 24$ and $x+2y+3z \leqslant 12$ are the inequalities which are binding at $(0,3,2)$, we define a valid inequality by adding the corresponding values in the two inequalities and subtracting 1 from the right-hand-side to get $4x+8y+6z \leqslant 35$. Then append this inequality to the (4.10) and go to Step 1 of the second iteration to improve the bound.

Iteration 2: $UB_1 = -16$, $LB_1 = -22$, and set the counter to $t = 2$.

Step 1: Re-optimize (4.10) again by appending the *valid inequality* $4x+8y+6z \leqslant 35$, i.e., solve the problem:

$$\begin{aligned} \min\ & F_1(x,y,z) = -xy - 4y - 5y \\ \text{s.t. } & 3x + 6y + 3z \leqslant 24 \\ & x + 2y + 3z \leqslant 12 \\ & 3x + 2y + z \geqslant 6 \\ & 4x + 8y + 6z \leqslant 35 \\ & (x,y,z) \in \mathbb{R}^1_+ \times \mathbb{R}^1_+ \times \mathbb{R}^1_+, \end{aligned} \tag{4.11}$$

using any linear programming solver to get an optimal solution $(\bar{x}^2, \bar{y}^2, \bar{z}^2) = (1,1,3)$ with the corresponding objective value of $F_1 = -20$. Note that this solution satisfies conditions (C1) and (C2), hence we update $LB_2 = F_1 = -20$ and go to the next step (**Step 2**).

The branching steps can be represented graphically in the enumeration tree shown in Figure 2.

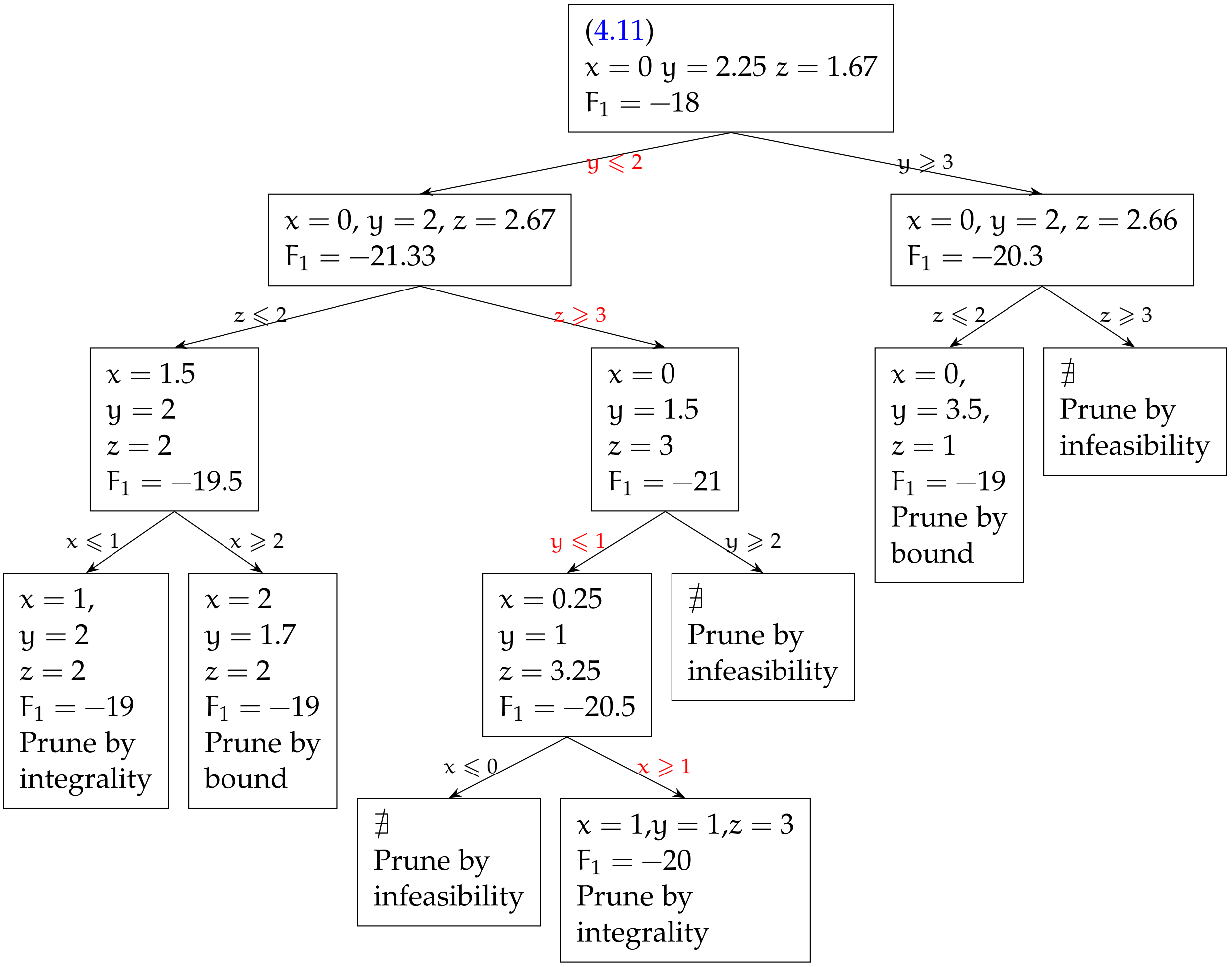

Figure 2: Branching steps in the enumeration tree of Example 4.2.

Step 2: Fix $x \longleftarrow 1$ and add the branching constraints $y \leqslant 2, z \geqslant 3, y \leqslant 1, x \geqslant 1$ (as shown in Figure 2) in the middle level of the updated problem and solve the relaxed problem

$$\begin{aligned} \min\ & F_2(x,y,z) = -2z \\ \text{s.t. } & 3x + 6y + 3z \leqslant 24 \\ & x + 2y + 3z \leqslant 12 \\ & 3x + 2y + z \geqslant 6 \\ & 4x + 8y + 6z \leqslant 35 \\ & y \leqslant 2, z \geqslant 3, y \leqslant 1, x \geqslant 1, x = 1 \\ & (x,y,z) \in \mathbb{R}^1_+ \times \mathbb{R}^1_+ \times \mathbb{R}^1_+ \end{aligned} \tag{4.12}$$

using any linear programming solver and get an integer solution $(\bar{y}^4, \bar{z}^4) = (0, 3)$. Since

$$F_2(\bar{x}^3, \bar{y}^3, \bar{z}^3) = -6 = F_2(\bar{x}^3, \bar{y}^4, \bar{z}^4),$$

the solution $(\bar{x}^3, \bar{y}^3, \bar{z}^3)$ is also optimal for (4.12) and hence we conclude that $(\bar{x}^3, \bar{y}^3, \bar{z}^3) = (1, 1, 3)$ is feasible for the second level problem. That means, it satisfies condition (C3). So, we update $UB_1 = F_1 = -20$ and go to the next step.

Step 3: Fixing $(x, y) \longleftarrow (1, 1)$ in the lower level problem we obtain the following problem.

$$\begin{aligned}
\min_{x_1} \ & F_3(x, y, z) = z \\
\text{s.t. } & 3x + 6y + 3z \leqslant 24 \\
& x + 2y + 3z \leqslant 12 \\
& 3x + 2y + z \geqslant 6 \\
& 4x + 8y + 6z \leqslant 35 \\
& y \leqslant 2, z \geqslant 3, y \leqslant 1, x \geqslant 1 \\
& x = 1, y = 1 \\
& (x, y, z) \in \mathbb{R}^1_+ \times \mathbb{R}^1_+ \times \mathbb{R}^1_+.
\end{aligned}$$

Solving this problem using any linear programming solver we get $\bar{z}^5 = 3$. Then, $(\bar{x}^5, \bar{y}^5, \bar{z}^5) = (1, 1, 3)$ satisfies condition (C4). Therefore, we set $UB_1 = -20 = LB_1$, and hence the algorithm terminates. That means, the point $(x^*, y^*, z^*) = (1, 1, 3)$ is the optimal solution of the problem presented in Eq. (4.9).

The following third example has the same structure as in Example 4.2, but contains more number of variables at each decision level.

Example 4.3. Consider the following tri-level integer linear programming problem:

$$\begin{aligned}
\min_{x_1, x_2} \ & F_1 = 9x_1 + 6x_2 + 6x_3 + 2x_4 + x_5 - 4x_6 \\
\text{s.t. } & \min_{x_3, x_4} F_2 = x_1 + 2x_3 - x_6 \\
& \quad \text{s.t. } \min_{x_5, x_6} \quad F_3 = x_4 + x_5 - x_6 \\
& \qquad \text{s.t. } 3x_1 - 3x_2 - 8x_3 + 10x_4 + 8x_5 + 5x_6 \leqslant 8 \\
& \qquad\quad 6x_1 + 4x_2 + 3x_3 + 10x_4 + 8x_5 - 7x_6 \leqslant 9 \\
& \qquad\quad x_1 + x_2 + x_3 + x_4 + x_5 - x_6 \leqslant 10 \\
& \qquad\quad -x_1 - x_2 - x_3 + x_4 - x_5 + x_6 \leqslant 4 \\
& \qquad\quad x_1 + x_2 + x_3 \leqslant 10 \\
& \qquad\quad x_3 + x_5 + x_6 \leqslant 8 \\
& \qquad x_1, x_2, x_3, x_4, x_5, x_6 \in \mathbb{Z}^6_+.
\end{aligned}$$

Applying the same procedures as in Example 4.2 above we can get the optimal solution of the problem to be $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (0, 0, 3, 0, 0, 6)$ with the corresponding objective values: $F_1 = F_2 = F_3 = -6$. □

5. Conclusion and further study

In this study, tri-level linear programming problems are considered, where all the involved variables are assumed to be discrete or integers. The paper introduces a simple and novel branch-and-cut algorithm for the global solution of such problems. The algorithm assumes that the relaxed constraint region is bounded and the reaction sets for each of the followers is nonempty corresponding to a given choice of leaders decision variable. The algorithm utilizes a modified branch-and-bound procedures together with the introduction of valid inequalities (or intersection cuts) to separate an infeasible point from the rest of the constraint region. Selected numerical examples are presented to demonstrate the performance of the algorithm. The results of the numerical examples show that the algorithm can solve various types of problems and it is very much promising.

One of the advantages of the proposed algorithm is that it employs any existing linear programming solver at each iteration. In addition, it can be easily extendable to solve multi-level discrete linear programming problems by recursively applying the same given procedures. However, the possibility of extending the same algorithm to solve mixed-integer linear multi-level programming problems can be explored in future works. In addition, the definition and characterization of valid inequalities or intersection cuts plays a crucial role in the performance of the Branch-and-cut algorithm for multilevel integer linear programming problems. Therefore, interested researchers can consider derivation of strong valid inequalities (or intersection cuts), especially for problems with more than two levels as possible future direction of research.

Acknowledgment

The research of the first author was supported by the International Science Program (ISP) of Sweden, through a research project at the Department of Mathematics, Addis Ababa University.

References

[1] N. Alguacil, A. Delgadillo, J. M. Arroyo, *A tri-level programming approach for electric grid defense planning*, Comput. Oper. Res., **41** (2014), 282–290. 1

[2] S. Avraamidou, E. N. Pistikopoulos, *Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems*, J. Global Optim., **74** (2019), 443–465. 1, 4

[3] J. F. Bard, *An investigation of the linear three level programming problem*, IEEE Trans. Systems Man Cybernet., **14** (1984), 711–717. 1, 2

[4] J. F. Bard, *Optimality conditions for the bi-level programming problem*, Naval Res. Log. Quar., **31** (1984), 13–26.

[5] G. Cornuéjols, *Valid inequalities for mixed integer linear programs*, Math. Program., **112** (2008), 3–44. 3.6

[6] S. Dempe, F. M. Kue, *Solving discrete linear bilevel optimization problems using the optimal value reformulation*, J. Glob. Optim., **68** (2017), 255–277. 3

[7] S. T. DeNegre, T. K. Ralphs, *A Branch-and-cut Algorithm for Integer Bilevel Linear Programs*, In: Operations Research and Cyber-Infrastructure. Operations Research/Computer Science Interfaces (Springer, Boston), **47** (2009), 65–78. 1, 3, 3.1, 3.6, 3.6

[8] N. P. Faísca, M. P. Saraiva, B. Rustem, N. E. Pistikopoulos, *A multiparametric programming approach for multilevel hierarchical and decentralized optimization problems*, Comput. Manag. Sci., **6** (2009), 377–397. 1

[9] M. Fischetti, I. Ljubić, M. Monaci, M. Sinnl, *Intersection Cuts for Bilevel Optimization*, In: Integer programming and combinatorial optimization (Springer, Switzerland), **2016** (2016), 77–88. 3.6

[10] M. Fischetti, I. Ljubić, M. Monaci, M. Sinnl, *A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs*, Oper. Res., **65** (2017), 1615–1637. 3, 3.2, 3.6

[11] C. Florensa, P. Garcia-Herreros, P. Misra, E. Arslan, S. Mehta, I. E. Grossmann, *Capacity planning with competitive decision-makers: Trilevel MILP formulation, degeneracy, and solution approaches*, European J. Oper. Res., **262** (2017), 449–463. 1

[12] T. R. Kannan, G. Dinakaran, N. J. Lavanya, *A Graphical Approach for Solving Three Variable Linear Programming Problems*, Recent Developments in Materials Processing (RDMP–04), Bharathiyar College of Engineering and Technology, Karaikal, India, (2004). 1

[13] A. M. Kassa, S. M. Kassa, *A multi-parametric programming algorithm for special classes of non-convex multilevel optimization problems*, Int. J. Optim. Control. Theor. Appl. IJOCTA, **3** (2013), 133–144. 1

[14] A. M. Kassa, S. M. Kassa, *A branch-and-bound multi-parametric programming approach for non-convex multilevel optimization with polyhedral constraints*, J. Glob. Optim., **64** (2016), 745–764. 1

[15] G. Y. Ke, J. H. Bookbinder, *Coordinating the discount policies for retailer, wholesaler, and less-than-truckload carrier under price-sensitive demand: A tri-level optimization approach*, Int. J. Pro. Econ., **196** (2018), 82–100. 1

[16] K. Lachhwani, A. Dwivedi, *Bi-level and Multi-Level Programming Problems: Taxonomy of Literature Review and Research Issues*, Arch. Comput. Methods Eng., **25** (2018), 847–877. 1

[17] J. V. Outrata, *On the numerical solution of a class of Stackelberg problems*, Z. Oper. Res., **34** (1990), 255–277. 3.2

[18] A. S. Safaei, S. Farsad, M. M. Paydar, *Robust bi-level optimization of relief logistics operations*, Appl. Math. Model., **56** (2018), 359–380. 1

[19] G. K. Saharidis, M. G. Ierapetritou, *Resolution method for mixed integer bi-level linear problems based on decomposition technique*, J. Global Optim., **44** (2009), 29–51. 3

[20] M. Sakawa, T. Matsui, *Interactive fuzzy stochastic multi-level 0–1 programming using tabu search and probability maximization*, Expert Syst. Appl., **41** (2014), 2957–2963. 1

[21] M. Sakawa, I. Nishizaki, M. Hitaka, *Interactive fuzzy programming for multi-level 0–1 programming problems through genetic algorithms*, Eur. J. Oper. Res., **114** (1999), 580–588. 1

[22] S. S. Sana, *A production-inventory model of imperfect quality products in a three-layer supply chain*, Decis. Support Syst., **50** (2011), 539–547. 1

[23] S. Tahernejad, T. K. Ralphs, S. T. DeNegre, *A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation*, Math. Program. Comput., **12** (2020), 529–568. 3

[24] U.-P. Wen, *Mathematical methods for multilevel linear programming*, Ph.D. Dissertation, Dept. Industrial Engineering, State University of New York, Buffalo (1981). 3.1

[25] U.-P. Wen, W. F. Bialas, *The hybrid algorithm for solving the three-level linear programming problem*, Comput. Oper. Res., **13** (1986), 367–377. 1, 3.1

[26] D. J. White, *Penalty function approach to linear trilevel programming*, J. Optim. Theory Appl., **93** (1997), 183–197. 1

[27] A. T. Woldemariam, S. M. Kassa, *Systematic evolutionary algorithm for general multilevel Stackelberg problems with bounded decision variables (SEAMSP)*, Ann. Oper. Res., **229** (2015), 771–790. 2

[28] X. Xu, Z. Meng, R. Shen, *A tri-level programming model based on conditional value-at-risk for three-stage supply chain management*, Comput. Ind. Eng., **66** (2013), 470–475. 1

[29] P. Xu, L. Wang, *An exact algorithm for the bi-level mixed integer linear programming problem under three simplifying assumptions*, Comput. Oper. Res., **41** (2014), 309–318. 3

[30] Y. Yao, T. Edmunds, D. Papageorgiou, R. Alvarez, *Tri-level optimization in power network defense*, IEEE Trans. Syst. Man Cybern., **37** (2007), 712–718. 1

[31] G. Zhang, J. Lu, J. Montero, Y. Zeng, *Model, solution concept, and Kth-best algorithm for linear trilevel programming*, Inf. Sci., **180** (2010), 481–492. 1