Elsevier Editorial System(tm) for Information Sciences
Manuscript Draft

Manuscript Number:

Title: Model, Solution Concept and the Kth-Best Algorithm for Linear Tri-level Programming

Article Type: Spec.Iss.: Modelling Uncertainty

Keywords: Multi-level programming, tri-level programming, bi-level programming, hierarchical decision-making, Kth-best algorithm, optimization

Corresponding Author: Dr jie lu, PhD

Corresponding Author's Institution: University of Technology, Sydney

First Author: jie lu, Phd

Order of Authors: jie lu, Phd; guangquan zhang, Phd; jie lu, PhD

Suggested Reviewers: Zhong Li Phd
A/professor
zhong.li@FernUni-Hagen.de

xianyi zeng Phd
Professor
xianyi.zeng@ensait.fr

peijun Guo Phd
A/professor
guo@ynu.ac.jp

Cover Letter an Original Article Statement

The paper we are herewith submitting is entitled “Model, Solution Concept and the *K*th-Best Algorithm for Linear Tri-level Programming” by Guangquan Zhang, Jie Lu, Javier Montero, and Yi Zeng.

This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere. All authors have checked the manuscript and have agreed to the submission.

Guangquan Zhang, Jie Lu, Javier Montero, and Yi Zeng

Model, Solution Concept and the *K*th-Best Algorithm for Linear Tri-level Programming

GUANGQUAN ZHANG*, JIE LU*, JAVIER MONTERO#, YI ZENG*

* *Faculty of Information Technology, University of Technology Sydney, Australia*
Email: {zhangg,jielu,yizeng@it.uts.edu.au}
Faculty of Mathematics, Complutense University of Madrid, Spain
Email: <Javier_Montero@mat.ucm.es>

Abstract
Tri-level programming deals with hierarchical optimization problems in which the top-level decision maker attempts to optimize his or her objective, but subjects it to a set of constraints and the reactions of decision makers at the middle-level and bottom-level. The middle-level's decision is at the same time influenced by the reactions of the bottom decision maker. To describe and solve this complex issue, this paper first proposes a tri-level programming model and a set of linear tri-level programming (LTLP) solution concepts. It then presents a *K*th-best algorithm to solve the linear tri-level programming problems. Finally, a case-based example further illustrates the proposed model, solution concept and algorithm.

Keywords: Multi-level programming, tri-level programming, bi-level programming, hierarchical decision-making, *K*th-best algorithm, optimization.

1. Introduction

Many organizational decisions are made in a multi-level hierarchical order and each decision level has no direct control upon the decision of the others, but actions taken by one decision level affect those from the others [3, 5]. Decision makers at all levels attempt to optimize their individuals' objectives, but the decisions are affected by the optimal objective values of the other levels. Such a hierarchical decision process appears naturally in many fields including decentralized resource planning [26], environmental policy [2], highway pricing [20], power market [19], logistics [35], economic [1], manufacturing [15] and road network management [17]. These kinds of decision problems are called multi-level decision problems or multi-level optimization problems [5, 13]. Multi-level programming typically solves the problems [3]. The literature shows that research on multi-level programming has mainly centered on the bi-level situation, called bi-level programming [4, 9, 14], and a linear version of the problem, called linear multi-level programming.

Bi-level programming (BLP) was proposed to deal with multi-level decision problems when only two decision levels are involved. The decision maker at the upper level is termed the leader, and at the lower level, the follower [16]. There have been many approaches and algorithms proposed for solving BLP problems since the field caught the attention of researchers in the mid-1970s, including the well-known Kuhn-Tucker approach [6, 18], the Branch-and-bound algorithm [8, 18], penalty function approach [34], the *K*th-best approach [13, 10], and also genetic algorithm [12, 22]. Furthermore, some fuzzy BLP models and approaches [29, 27, 35], multi-follower BLP [23, 24, 25], and multi-objective BLP models and approaches [17, 28, 30, 36] have been recently developed to deal with more complex cases of bi-level decision problems.

Compared with bi-level decision-making, more organizational decision problems, which require compromises among the objectives of several interacting decision entities, are allocated in a tri-level hierarchy. The execution of decisions is sequential, from top to middle and then to bottom levels. Each entity in the tri-level hierarchy independently maximizes/minimizes its own objectives, but is affected by the actions of the other two decision entities through externalities. We use a university example to explain the kind of problem. A university aims to improve its research quality through new research development strategies. The strategies made at university level directly affect the research strategy-making in its faculties. This process continues within a hierarchy of decision entities to its departments. In the meantime, the actions at the faculty level may affect the research development strategies sought by the university and the actions at the department level may affect its faculty's. Each related decision entity in this university wishes to optimize its individual research development objective (such as more research outcomes) in view of the partial control exercised at other levels. The university's decision makers can control the effect by exercising preemptive partial control over the university through budget modifications or regulations, but subject to a set of constraints (such as limited research funding and working load), and possible reactions from its faculties and also departments. Obviously, the complexity of decision problems increases significantly when the number of levels is greater than two [11]. As a tri-level decision reflects the main features of multi-level decision problems, the models and methods developed for tri-level decisions can be easily extended to other multi-level decision problems (the number of levels>3).

Tri-level programming has been studied by some researchers and the results have mainly focused on its linear version. For example, Bard and Falk [7] first proposed the necessary conditions for the linear three-level programming problem based on Stackelberg game theory, and then developed rational reaction sets for each of the decision entities and a cutting plane algorithm to solve the LTLP case. White [33] proposed a penalty function approach to the LTLP problems. In this approach, each decision-maker had an objective function to be optimized within the imposed constraint set. The top-level decision-maker selects an action first, within a specified constraint set, then the middle-level decision-maker selects an action within a constraint set determined by the action of the top-level decision-maker, and finally the bottom-level decision-maker selects an action within a constraint set determined by the actions of decision-makers at the top-level and middle-level. Obviously, this approach may lead to a paradox that lower level decision power dominates upper level decisions. Lai [21] applied the concepts of memberships of optimality and degrees of decision powers to solve the technical inefficiency problems in Kuhn-Tucker conditions or penalty functions-based traditional tri-level programming approaches. In this method, the leader first sets memberships of optimality of their possible objective values and decisions, as well as their decision power; and then asks the follower for their optima calculated in isolation under given constraints. The follower's decision, with the corresponding levels of optimality and decision powers, is submitted to and modified by the leader with considerations of overall benefit for the organization and distribution of decision power until a best preferred solution is reached. Subsequently, Sinha and Sinha [30] proposed a Karush-Kuhn-Tucker (KKT) transformation method for multi-level linear programming problems, wherein some subsets of decision variables were under the exclusive domain of the decision makers as some other subsets of decision variables are also common to two or more decision makers on different levels and/or on various divisions on a level. Although these LTLP methods and algorithms have been developed, the solution concepts, including the solution existence for LTLP problems, have not been well developed in the literature. Also, in almost all exiting LTLP models, decision entities' constraints were written at the same level rather than in separate decision levels. The popular Kuhn-Tucker approach and related Branch-and-bound algorithm, as well as penalty function approach, have been well extended from bi-level programming problems to tri-level programming problems; the popular *K*th-best approach should also be extended for dealing with LTLP problems.

Our study addresses the complex tri-level programming problem in its linear version. To solve the LTLP problem, we first propose a related solution concept and solution existence theorem. We then develop a tri-level *K*th-best algorithm based on the solution concept to obtain a solution. Following the introduction, Section 2 introduces a LTLP model. The solution concepts of the LTLP model and related theorems and corollaries are presented in Section 3. Section 4 proposes a linear tri-level *K*th-best algorithm for solving the LTLP problems. Section 5 illustrates the proposed LTLP model, the solution concept, and the linear tri-level *K*th-best algorithm using a case-based numerical example. Section 6 concludes this paper and highlights our further study.

2. Linear Bi-Level and Tri-Level Programming Models

This section will introduce the models of bi-level programming and tri-level programming.

2.1 A bi-level programming model

A bi-level decision problem can be viewed as a static version of the non-cooperative, two-player (decision entity) game [31, 32]. The decision entity at the upper level is termed as leader, and the lower level, follower. The control for the decision variables is partitioned amongst the decision entities who seek to optimize their individual objective function [5].

Bi-level programming typically models bi-level decision problems, in which the objectives and the constraints of both upper and lower level decision entities (leader and follower) are expressed by linear or nonlinear functions, as follows:

For $x \in X \subset R^n$, $y \in Y \subset R^m$, $F : X \times Y \to R^1$, and $f : X \times Y \to R^1$,

$$\begin{aligned}
&\min_{x\in X} F(x,y)\\
&\text{subject to } G(x,y) \le 0\\
&\qquad\qquad \min_{y\in Y} f(x,y)\\
&\qquad\qquad \text{subject to } g(x,y) \le 0,
\end{aligned}$$

where the variables x, y are called the leader and the follower variable respectively and $F(x, y)$ and $f(x, y)$ the leader's and the follower's objective functions.

This model aims to find a solution to the upper level problem $\min_{x\in X} F(x,y)$ subject to its constraints $G(x,y) \le 0$ where, for each value of leader's variable x, y is the solution of the lower level problem $\min_{y\in Y} f(x,y)$ under its constraints $g(x,y) \le 0$.

2.2 A tri-level programming model

In a tri-level hierarchical decision problem, each decision entity at one level has its objective determined and variables, in part, controlled by the entities at other levels. The choice of values for its variables may allow it to influence the decisions made at other levels, and thereby improve its own objective. To describe a tri-level decision problem, a basic LTLP model is described as follows:

For $x \in X \subset R^n$, $y \in Y \subset R^m$, $z \in Z \subset R^p$, $f_i : X \times Y \times Z \to R^1$, $i = 1,2,3,$

$$\begin{aligned}
&\min_{x\in X} \ f_1(x,y,z) = \alpha_1 x + \beta_1 y + \mu_1 z \\
&\text{subject to } \ A_1 x + B_1 y + C_1 z \le b_1 \qquad\qquad (1) \\
&\qquad \min_{y\in Y} \ f_2(x,y,z) = \alpha_2 x + \beta_2 y + \mu_2 z \\
&\qquad \text{subject to } \ A_2 x + B_2 y + C_2 z \le b_2 \\
&\qquad\qquad \min_{z\in Z} \ f_3(x,y,z) = \alpha_3 x + \beta_3 y + \mu_3 z \\
&\qquad\qquad \text{subject to } \ A_3 x + B_3 y + C_3 z \le b_2
\end{aligned}$$

where $\alpha_i \in R^n$, $\beta_i \in R^m$, $\mu_i \in R^p$, $b_i \in R^{q_i}$, $A_i \in R^{q_i \times n}$, $B_i \in R^{q_i \times m}$, $C_i \in R^{q_i \times p}$, $i = 1,2,3.$ The variables x, y, z are called the top-level, middle-level, and bottom-level variables respectively, and $f_1(x, y, z)$ ($f_2(x, y, z)$, $f_3(x, y, z)$) the top-level, middle-level and bottom-level objective functions. In this model, we can see that the decision problem has three optimization sub-problems (represented by three objective functions respectively) in a three-level hierarchy of decisions. Each level has individual control variables within its optimization sub-problem, but also takes other levels' variables in its optimization sub-problem.

3. Solution Concepts for Linear Tri-Level Programming

To obtain an optimized solution to this LTLP problem, we first propose its solution definition as follows:

Definition 3.1

(1) Constraint region of the LTLP:

$$S = \{(x, y, z) : x \in X, y \in Y, z \in Z, A_i x + B_i y + C_i z \le b_i, i = 1, 2, 3\}$$

(2) Feasible set for the middle and bottom levels for each fixed $x \in X$:

$$S(x) = \{(y,z) \in Y \times Z : B_i y + C_i z \le b_i - A_i x, i = 2, 3\}$$

(3) Feasible set for the bottom level for each fixed $(x, y) \in X \times Y$:

$$S(x, y) = \{z \in Z : C_3 z \le b_3 - A_3 x - B_3 y\}$$

(4) Projection of S onto the top level's decision space:

$$S(X) = \{x \in X : \exists (y, z) \in Y \times Z,\ A_i x + B_i y + C_i z \le b_i,\ i = 1, 2, 3\}$$

(5) Projection of S onto the top and middle level's decision space:

$$S(X, Y) = \{(x, y) \in X \times Y : \exists z \in Z,\ A_i x + B_i y + C_i z \le b_3,\ i = 1, 2, 3\}$$

(6) The rational reaction set of the middle and bottom levels for $x \in S(X)$:

$$P(x) = \{(y, z) : (y, z) \in \arg\min[f_2(x, \hat{y}, \hat{z}) : (\hat{y}, \hat{z}) \in S(x), \hat{z} \in \arg\min[f_3(x, \hat{y}, \tilde{z}) : \tilde{z} \in S(x, \hat{y})]\}.$$

(7) The rational reaction set of the bottom level for $(x, y) \in S(X, Y)$:

$$P(x, y) = \{z : z \in \arg\min[f_3(x, y, \hat{z})] : \hat{z} \in S(x, y)\}$$

(8) Inducible region:

$$IR = \{(x, y, z) : (x, y, z) \in S, (y, z) \in P(x)\}$$

So, finding the solution of (1) is equal to solve

$$\min\{f_1(x, y, z) : (x, y, z) \in IR\}$$

We give the three assumptions below in order to introduce the solution existence theorem.

Assumptions

(1) S is nonempty and compact.

(2) For decisions taken by the leader, the follower has some rooms to respond; i.e, $P(x) \ne \phi$ and $P(x, y) \ne \phi$.

(3) $P(x)$ and $P(x, y)$ are point to point maps with respect to x and (x, y) respectively.

We proposed three important LTLP theorems below. Theorem 3.1 proves the existence of an optimal solution to the LTLP model. Theorem 3.2 presents a way to achieve a solution of the LTLP problem. To develop the *K*th-best algorithm, Theorem 3.3 provides the necessary foundations.

Theorem 3.1. If S is nonempty and compact, then there exists an optimal solution for a LTLP problem.

Proof. Since S is non-empty, suppose $(x^*, y^*, z^*) \in S$. Then we have $S(x^*)$ is non-empty. Thus

$$P(x^*) = \{(y, z) : (y, z) \in \arg\min[f_2(x^*, \hat{y}, \hat{z}) : (\hat{y}, \hat{z}) \in S(x^*), \hat{z} \in \arg\min[f_3(x, \hat{y}, \tilde{z}) : \tilde{z} \in S(x^*, \hat{y})]\}. \quad (2)$$

Let

$$Q(x^*) = \min[f_2(x^*, \hat{y}, \hat{z}) : (\hat{y}, \hat{z}) \in S(x^*), \hat{z} \in \arg\min[f_3(x, \hat{y}, \tilde{z}) : \tilde{z} \in S(x^*, \hat{y})]] \quad (3)$$

$$Q(x^*, \hat{y}) = \min[f_3(x, \hat{y}, \tilde{z}) : \tilde{z} \in S(x^*, \hat{y})] \quad (4)$$

Notice that for any given x^* and $\hat{y}$, $Q(x^*, \hat{y})$ is non-empty. Hence $Q(x^*)$ is non-empty and, in turn, $P(x^*)$ is non-empty. For any $(y, z) \in P(x^*)$, it is known that (y, z) is an optimal solution of the middle and bottom level programming problem under given x^*. Hence, (x^*, y, z) is a feasible solution of the LTLP.

Theorem 3.2. The inducible region can be written equivalently as a piecewise linear equality constraint comprised of support hyper-planes of *S*.

Proof. To see this, let us use the notations in the proof of Theorem 3.1 to rewrite the inducible region as follows:

$$IR = \{(x, y, z) \in S;\ \beta_2 y + \mu_2 z = \min[\beta_2 \hat{y} + \mu_2 \hat{z};\ B_i \hat{y} + C_i \hat{z} \le b_i - A_i x,\ \hat{y} \ge 0, \hat{z} \ge 0, i = 2,3],$$
$$\mu_3 z = \min[\mu_3 \tilde{z};\ C_3 \tilde{z} \le b_3 - A_3 x - B_3 y +,\ \tilde{z} \ge 0]\} \quad (5)$$

It needs to prove that $Q(x)$ is a piecewise linear equality constraint.

According to the expressions of $Q(x)$ and $Q(x, y)$, we first prove that $Q(x, y)$ is a piecewise linear equality constraint for any given *x* and *y*. Because $Q(x, y)$ can be seen as a linear programming problem with parameters *x* and *y*, the dual problem of $Q(x, y)$ is

$$\max\{u(b_3 - Ax_3 - By_3) : uC_3 \ge -\mu_3, u \ge 0\}. \quad (6)$$

The problem has the same optimal values as $Q(x, y)$ at the solution u^*. Let $u^1, \cdots, u^t$ be a listing of all the vertices of the constraint region of the dual problem given by $U = \{u : uC_3 \ge -\mu_3\}$. As a solution of the dual problem occurs at a vertex of U, we have the equivalent problem

$$\max\{u(b_3 - Ax_3 - By_3) : u \in \{u^1, \cdots, u^t\}\}. \quad (7)$$

This means that $Q(x, y)$ is a piecewise linear function.

Next we will prove that $Q(x)$ is a piecewise linear function. Suppose $z^1, z^2, \cdots, z^s$ are solutions for problem $Q(x, y)$. For each z^i, $Q(x)$ becomes a programming problem with parameters x and z^i. Therefore we have *s* parameterized programming problems $Q(x)|_{z^1}, \cdots, Q(x)|_{z^s}$, respectively. Similarly, each $Q(x)|_{z^i}$ is a piecewise linear function. Hence, IR can be rewritten as

$$IR = \bigcup_{i=1}^{s} \{(x, y, z^i) : \beta_2 y = Q(x)|_{z^i} - \mu_2 z^i\} \quad (8)$$

and is a piecewise linear equality constraint.

Corollary 3.1 A solution to the LTLP problem (1) occurs at a vertex of *IR*.

Theorem 3.3. The solution (*x**, *y**, *z**) of the linear tri-level programming problem occurs at a vertex of *S*.

Proof. Let $(x^1, y^1, z^1), \cdots, (x^t, y^t, z^t)$ be the distinct vertices of *S*. Since any point in *S* can be written as a convex combination of these vertices, let $(x^*, y^*, z^*) = \sum_{i=1}^{\bar{t}} \delta_i (x^i, y^i, z^i)$, where $\sum_{i=1}^{\bar{t}} \delta_i = 1, \delta_i \ge 0, i = 1, \cdots, \bar{t}$ and $\bar{t} \le t$. It must be shown that $\bar{t} = 1$. To see this, let us write the constraints of (1) at (*x**, *y**, *z**) in their piecewise linear form (8).

$$0 = Q(x^*)|_{z^*} - \beta_2 y^* - \mu_2 z^*$$
$$= Q(\sum_{i=1}^{\bar{t}} \delta_i x^i)|_{z^*} - \beta_2 \left(\sum_{i=1}^{\bar{t}} \delta_i y^i \right) - \mu_2 \left(\sum_{i=1}^{\bar{t}} \delta_i z^i \right)$$

$$\leq \sum_{i=1}^{\bar{t}} \delta_i Q(x^i)|_{z^*} - \sum_{i=1}^{\bar{t}} \delta_i \beta_2 y^i - \sum_{i=1}^{\bar{t}} \delta_i \mu_2 z^i$$
$$= \sum_{i=1}^{\bar{t}} \delta_i (Q(x^i)|_{z^*} - \beta_2 y^i - \mu_2 z^i)$$

by convexity of $Q(x)$. But by definition,

$$Q(x^i)|_{z^*} = \min_{\substack{(y,z)\in S(x^i) \\ z\in p(x^i,y)}} (\beta_2 y + \mu_2 z) \leq \beta_2 y^i + \mu_2 z^i.$$

Therefore, $Q(x^i)|_{z^*} - \beta_2 y^i - \mu_2 z^i \leq 0, i = 1, \cdots, \bar{t}$. Noting that $\delta_i > 0, i = 1, \cdots, \bar{t}$, the equality in the preceding expression must hold or else a contradiction would result in the sequence above. Consequently, we have $Q(x^i)|_{z^*} - \beta_2 y^i - \mu_2 z^i = 0$ for all i. These imply that $(x^i, y^i, z^i) \in IR$, $i = 1,\ldots, \bar{t}$. and that (x^*, y^*, z^*) can be written as a convex combination of points in IR. Because (x^*, y^*, z^*) is a vertex of IR by Corollary 1 and $P(x)$ and $P(x, y)$ are single-valued, a contradiction results unless $\bar{t} = 1$.

Corollary 3.2 If (x, y, z) is an extreme point of IR, then it is an extreme point of S.

The above theorems and corollaries provide a theoretical foundation and a suitable way to solve the proposed LTLP problem. We can therefore only search extreme points on the constraint region S to find an optimal solution for a LTLP problem.

This Corollary provides a clear way to obtain a solution for the LTLP problem by the *K*th-best algorithm.

4. The Linear Tri-Level *K*th-Best Algorithm

Based on the bi-level *K*th-best algorithm and Theorem 3.3 and Corollary 3.2, we propose a linear tri-level *K*th-best algorithm to solve the LTLP problem.

Suppose to solve the LTLP problem below:

$$\min\{\alpha_1 x + \beta_1 y + \mu_1 z : (x, y, z) \in S\} \tag{9}$$

The N ranked basic feasible solution to (9) is

$$(x_{[1]}, y_{[1]}, z_{[1]}), (x_{[2]}, y_{[2]}, z_{[2]}), \ldots, (x_{[N]}, y_{[N]}, z_{[N]}),$$

such that $\alpha_1 x_{[i]} + \beta_1 y_{[i]} + \mu_1 z_{[i]} \leqq \alpha_1 x_{[i+1]} + \beta_1 y_{[i+1]} + \mu_1 z_{[i+1]}$, $i = 1, \ldots, N-1$. Then solving this problem is equivalent to finding the index

$$K^* \triangleq \min\{i \in \{1, \ldots, N\} : (x_{[i]}, y_{[i]}, z_{[i]}) \in IR\}.$$

Therefore the global solution is $(x_{[K^*]}, y_{[K^*]}, z_{[K^*]})$. Similarly, for the (middle level, bottom level) problem:

$$\min\{\beta_2 y + \mu_2 z : (y, z) \in P(x_{[i]})\},$$

the ranked basic feasible solution becomes

$$(\tilde{y}_{[1]}, \tilde{z}_{[1]}), (\tilde{y}_{[2]}, \tilde{z}_{[2]}), \ldots, (\tilde{y}_{[N]}, \tilde{z}_{[N]}),$$

such that $\beta_2\tilde{y}_{[i]} + \mu_2\tilde{z}_{[i]} \leqq \beta_2\tilde{y}_{[i+1]} + \mu_2\tilde{z}_{[i+1]}$, $i = 1, ..., N-1$. Then solving this problem is equivalent to finding the index $K^{**} \triangleq \min\{i \in \{1, ..., N\} : (\tilde{y}_{[i]}, \tilde{z}_{[i]}) \in IR\}$ yielding the optimal solution for (middle level, bottom level) problem $(\tilde{y}_{[K^{**}]}, \tilde{z}_{[K^{**}]})$.

The procedure of the linear tri-level *K*th-best algorithm is described as follows:

Step 1: Put $i \leftarrow 1$. Solve problem (9) with the simplex method to obtain the optimal solution $(x_{[1]}, y_{[1]}, z_{[1]})$. Let $W = \{(x_{[1]}, y_{[1]}, z_{[1]})\}$ and $T = \phi$. Go to Step 2.

Step 2: Treat the problem as a top (middle bottom) level problem. This step turns into solving the follower (middle bottom)'s decision problem below with the bi-level *K*th-best algorithm to see if the examining extreme point is the optimal solution to problem (9).

$$\min\{\beta_2 y + \mu_2 z : (y, z) \in P(x_{[i]})\} \tag{10}$$

Let $(\tilde{y}, \tilde{z})$ denote the optimal solution to (10). If $\tilde{y} = y_{[i]}$ and $\tilde{z} = z_{[i]}$, stop and $(x_{[i]}, y_{[i]}, z_{[i]})$ is the global optimum to (9) with $K^* = i$; Otherwise, go to Step 3.

Step 3: Let $W_{[i]}$ denote the set of adjacent extreme points of $(x_{[i]}, y_{[i]}, z_{[i]})$ such that $(x, y, z) \in W_{[i]}$ implies $\alpha_1 x + \beta_1 y + \mu_1 z \geq \alpha_1 x_{[i]} + \beta_1 y_{[i]} + \mu_1 z_{[i]}$. Let $T = T \cup \{(x_{[i]}, y_{[i]}, z_{[i]})\}$ and $W = (W \cup W_{[i]}) \setminus T$. Go to Step 4.

Step 4: Set $i \leftarrow i + 1$ and choose $(x_{[i]}, y_{[i]}, z_{[i]})$ so that

$$\alpha_1 x_{[i]} + \beta_1 y_{[i]} + \mu_1 z_{[i]} = \min\{\alpha_1 x + \beta_1 y + \mu_1 z : (x, y, z) \in W\}$$

Go to Step 2.

[END]

Figure 1 explains the tri-level *K*th-best algorithm by a flow chart. The tri-level *K*th-best algorithm uses two other algorithms: the simplex algorithm which addresses the problem of getting an optimal solution for single-level linear programming and the algorithm for finding adjacent extreme points of a selected extreme point. According to the results given by Bard [5], an extreme point is a geometrical interpretation of a feasible solution. Hence enumerating the adjacent extreme points is equivalent to enumerating all the feasible solutions for the decision problem. In the chart, the adjacent extreme point set is $W_{[i]}$ in the linear tri-level *K*th-best algorithm.

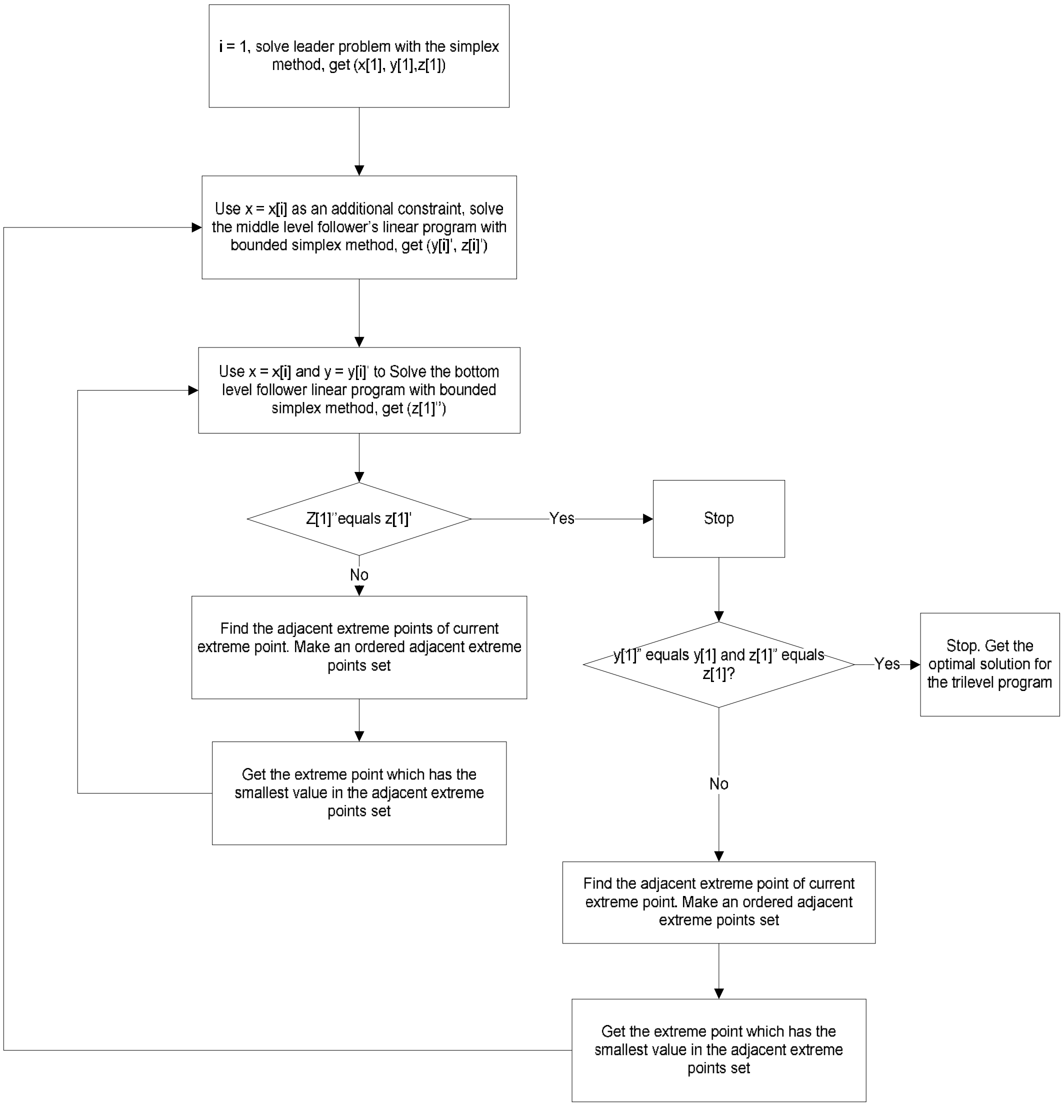

Figure 1: The Tri-level *K*th-best algorithm flow chart

5. An Illustrated Example

This section develops a case-based example on the annual budget decision-making of a company, where the CEO is the top-level decision entity, the heads of the company's branches act as the middle-level decision entity, and the group supervisors are the bottom-level decision entity. Obviously, the decision of the CEO will be affected by the reactions from the branch heads, and the decision of the branch heads tri-level by the reactions from the group supervisors. In order to arrive at an optimal solution (better strategy) for the company on the annual budget, we establish a tri-level programming model for this problem.

The CEO has its objective "to maximize the net profits", represented by min $f_1(x, y, z)$ (here $-f_1(x, y, z)$ is net profits), and x is the top-level decision control variable. The heads of the branches attempt "to minimize cost", represented by min $f_2(x, y, z)$, and y is the middle-level decision control variable. The bottom-level entity has their objective of "to maximize worker satisfactory degree", represented by min$f_3(x, y, z)$ (here, $-f_3(x, y, z)$ is worker satisfactory degree), and z is the bottom-level decision control variable. The objectives of the CEO, branch heads and group supervisors are subject to their particular constraints respectively such as material requirements, marking cost, labor cost, and working hours. This tri-level decision problem is briefly modeled by the following symbolical LTLP model:

For $x \in X, y \in Y, z \in Z, f_i : X \times Y \times Z \to R^1$,

$$\min_{x \in X} f_1 : x - 4y + 2z$$
Subject to: $-x - y \leqq -3$
$-3x + 2y - z \geqq -10$
$$\min_{y \in Y} f_2 : x + y - z$$
Subject to: $-2x + y - 2z \leqq -1$
$2x + y + 4z \leqq 14$
$$\min_{z \in Z} f_3 : x - 2y - 2z$$
Subject to: $2x - y - z \leqq 2$.

Now we use the developed linear tri-level *K*th-best algorithm to achieve a solution for this problem by the following steps.

Step 1: According to the tri-level *K*th-best algorithm, solving this problem requires first considering the middle level and the bottom level as a whole (middle, bottom), and then solving the problem using the bi-level *K*th-best algorithm. Each time we get a solution (x, y, z) from the outer level (top, (middle, bottom)) problem, we examine it by using the bi-level *K*th-best algorithm against the middle level and bottom level. Provided the optimal solution in the (middle, bottom) bi-Level problem is the same of (x, y, z) then we get the global optimal solution.

The problem then becomes to solve:

$$\min_{z \in Z} f_1 : x - 4y + 2z$$
Subject to: $-x - y \leqq -3$
$-3x + 2y - z \geqq -10$
$-2x + y - 2z \leqq -1$
$2x + y + 4z \leqq 14$
$2x - y - z \leqq 2$

First, solve the problem using simplex method, yielding $(3.75, 6.5, 0)$. Then put this solution into an extreme point set W so that $W = \{(3.75, 6.5, 0)\}$ and $T = \phi$, which is another extreme point set for storing the examined extreme points. Hence the x variable in this basic feasible solution is 3.75, use this as a new constraint of the middle level, the problem is to solve:

$$\min_{y \in Y} f_2 : x + y - z$$
Subject to: $-x - y \leqq -3$
$-3x + 2y - z \geqq -10$
$-2x + y - 2z \leqq -1$
$2x + y + 4z \leqq 14$

$$2x - y - z \leqq 2$$
$$x = 3.75 \quad (11)$$

Step 2: Use simplex method on the problem (11) and get a new feasible solution $(3.75, 5.17, 0.33)$. Then we need to use bi-level *K*th-best algorithm on the (middle, bottom) problem to test whether this is the optimal solution for the (middle, bottom) problem or not. To do that, let $x = 3.75, y = 5.17$, then the problem is equivalent to solving:

$$\min_{z \in Z} f_3 : x + y - z$$

Subject to: $-x - y \leqq -3$
$$-3x + 2y - z \geqq -10$$
$$-2x + y - 2z \leqq -1$$
$$2x + y + 4z \leqq 14$$
$$2x - y - z \leqq 2$$
$$x = 3.75$$
$$y = 5.17 \quad (12)$$

Step 3: Use simplex algorithm on (12). The result is different with the feasible solution of the selected adjacent extreme point. Consequently, it seeks to find the adjacent extreme points of the currently examined feasible solution (a $W[i]$ set), puts the previous selected adjacent extreme point into set T, then makes $W = (W \cup W[i]) \setminus T$, ranks the W set and retrieves the adjacent extreme point which has the smallest objective value from the new W. Take this adjacent extreme point as a new feasible solution and change the value of the y variable in (12) to the value of the y variable of this adjacent extreme point, continue Step 3. If the W in the (middle, bottom) problem is empty, keep the current solution, go to Step 5. Note that the T and W in this step are the extreme point sets in the (middle, bottom) problem.

Step 4: If we are not able to get a global solution after Step 3, we need to generate the corresponding T as well as W for the top (middle, bottom) problem, and get a new adjacent extreme point from the ranked adjacent extreme point-set W. This new adjacent extreme point will be used in Step 2, which continues processing the algorithm until the global solution, extreme point (4, 6, 0), is found. For the global solution, the objective value of f_1 is -20 and the objectives of f_2 and f_3 are 10 and -8, respectively. That is, the profit is 20, the cost is 10 and the worker satisfactory degree is 8. All results here are already rounded as an integer value for better understanding of the values.

Based on the algorithm, a software tool has been developed. After inputting all coefficients of all objectives and constrains of all the three decision entities, the software will use the tri-level *K*th-best algorithm to generate a solution to the tri-level decision problem.

6. Conclusions and Further Study

In a hierarchical organization, interactive decision entities exist within a predominantly hierarchical structure and the execution of decisions is sequential, from top to middle and then to bottom levels. Each entity independently maximizes its own objective, but is affected by the actions of other entities at different levels through externalities. To support this kind of complex decision-making, this paper proposes a solution definition and related theorems for LTLP problems. It also develops a tri-level *K*th-best algorithm to achieve a solution for LTLP problems.

As multiple followers commonly appear in both middle and bottom levels in a tri-level decision problem, our further study will include extending the tri-level *K*th-best algorithm to deal with multi-follower tri-level decision problems and will consider the various relationships among these followers at both middle-level and bottom-level. We will also improve the developed tri-level decision software to implement the multi-follower tri-level programming algorithm, and apply it in more real-world applications.

Acknowledgment

The work presented in this paper was supported by Australian Research Council (ARC) under discovery project grant DP0557154.

References

1. Amat J, McCarl B. A representation and economic interpretation of a two-level programming problem. Journal of the Operational Research Society 1981; 32; 783–792.
2. Amouzegar MA, Moshirvaziri K. Strategic management decision support system: An analysis of the environmental policy issues. Environmental Modeling and Assessment 2001; 6; 297-306.
3. Anandalingam G, Friesz T. Hierarchical optimization: An introduction. Annals of Operations Research 1992; 34; 1-11.
4. Bard J. Optimality conditions for the bilevel programming problem. Naval Research Logistics Quarterly 1984; 31; 13-26.
5. Bard J. Practical bilevel optimization Kluwer Academic Publishers: Dordrecht; 1998.
6. Bard J, Falk J. An explicit solution to the multi-level programming problem. Computer and Operations Research 1982; 9; 77-100.
7. Bard J, Falk J. Necessary conditions for the linear three level programming problem. 21st IEEE Conference on Decision and Control 1982; 21; 642-646
8. Bard J, Moore JT. A branch and bound algorithm for the bilevel programming problem. SIAM Journal on Scientific and Statistical Computing 1990; 11; 281–292.
9. Ben-Ayed O. Bilevel linear programming. Computers and Operations Research 1993; 20; 485-501.
10. Bialas W, Karwan M. Two-level linear programming. Management Science 1984; 30; 1004-1020.
11. Blair C. The computational complexity of multi-level linear programs. Annals of Operations Research 1992; 34; 13-19.
12. Calvete HI, Gale C, Mateo PM. A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research 2008; 188; 14-28.
13. Candler W, Townsley R. A linear two-level programming problem. Computers and Operations Research 1982; 9; 59-76.
14. Cao D, Leung LC. A partial cooperation model for non-unique linear two-level decision problems. European Journal of Operational Research 2002; 140; 134-141.
15. Cao D, Chen M. Capacitated plant selection in a decentralized manufacturing environment: A bilevel optimization approach. European Journal of Operational Research 2006; 169; 97-110.

16. Dempe S. Foundations of bilevel programming. Kluwer Academic Publishers: Dordrecht; 2002.
17. Feng C, Wen C. Bi-level and multi-objective model to control traffic flow into the disaster area post earthquake. Journal of the Eastern Asia Society for Transportation Studies 2005; 6; 4253 – 4268.
18. Hansen P, Jaumard B, Savard G. New branch-and-bound rules for linear bilevel programming. SIAM Journal on Scientific and Statistical Computing 1992; 13; 1194-1217.
19. Hobbs BF, Metzler B, Pang JS. Strategic gaming analysis for electric power systems: An MPEC approach. IEEE Transactions on Power System 2000; 15; 637–645.
20. Labbé M, Marcotte P, Savard G. A bilevel model of taxation and its application to optimal highway pricing. Management Science 1999; 44; 1608-1822.
21. Lai YJ. Hierarchical optimization: A satisfactory solution. Fuzzy Sets and Systems 1996; 77; 321-335.
22. Liu B, Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. Computers & Mathematics with Applications, 1998; 36; 79-89.
23. Lu J, Shi C, Zhang G. On bilevel multi-follower decision-making: general framework and solutions. Information Sciences 2006; 176; 1607-1627.
24. Lu J, Shi C, Zhang G, Dillon T. Model and extended Kuhn-Tucker approach for bilevel multi-follower decision making in a referential-uncooperative situation. International Journal of Global Optimization 2007; 38; 597–608.
25. Lu J, Shi C, Zhang G, Ruan D. An extended branch and bound algorithm for bilevel multi-follower decision making in a referential-uncooperative situation. International Journal of Information Technology and Decision Making 2007; 6; 371–388.
26. Lukac Z, Soric K, Rosenzweig VV. Production planning problem with sequence dependent setups as a bilevel programming problem. European Journal of Operational Research 2008; 187; 1504-1512.
27. Sakawa M, Nishizaki I, Uemura Y. Interactive fuzzy programming for multilevel linear programming problems with fuzzy parameters, Fuzzy Sets and Systems 2000; 109; 3-19.
28. Shi X, Xia H. Interactive bilevel multi-objective decision making. Journal of the Operational Research Society 1997; 48; 943–949.
29. Shih HS, Lai YJ, Lee ES. Fuzzy approach for multi-level programming problems. Computers and Operations Research 1996; 23; 73-91
30. Sinha S, Sinha SB. KKT transformation approach for multi-objective multi-level linear programming problems. European Journal of Operational Research 2002; 143; 19-31.
31. Stackelberg HV. Theory of the market economy. Oxford University Press: NewYork; 1952.
32. Wen UP, Hsu ST. Linear bilevel programming problems - A review. Journal of the Operational Research Society 1991; 42; 123–133.
33. White D. Penalty function approach to linear trilevel programming. Journal of Optimization Theory and Applications 1997; 93; 183-197.
34. White D, Anandalingam G. A penalty function approach for solving bi-level linear programs. Journal of Global Optimization 1993; 3; 397-419.

35. Zhang G, Lu J. Model and approach of fuzzy bilevel decision making for logistics planning problem. Journal of Enterprise Information Management 2007; 20; 178-197.
36. Zhang G, Lu J, Dillon T. Decentralized multi-objective bilevel decision making with fuzzy demands. Knowledge-Based Systems 2007; 20; 495-507.