International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

Line Search and Genetic Approaches for Solving Linear Tri-level Programming Problem

Eghbal Hosseini[1]

Department of Mathematics, Payam Noor University of Tehran, Tehran, Iran

Isa Nakhai Kamalabadi

Department of Industry, University of Kurdistan, Sanandaj, Iran

Abstract

In the recent years, the multi-level programming problems specially the bi-level and tri-level programming problems (TLPP) are interested by many researchers and these problems, particularly TLPP, are known as an appropriate tool to solve the real problems in several areas of optimization such as economic, traffic, finance, management, transportation, computer science and so on. Also, it has been proven that the general bi-level and TLPP are NP-hard problems. The literature shows it has been proposed a few attempts for solving using TLPP. In this paper, we attempt to propose a new function for smoothing the tri-level programming problem after using Karush-Kuhn-Tucker condition, also we develop two effective approaches, one based on Genetic algorithm, which it is an approximate approach, and the other based on the hybrid algorithm by combining the proposed method based on penalty function and the line search algorithm for solving the linear TLPP. In both of these approaches, by using the Karush-Kuhn-Tucker condition the TLPP is converted to a non-smooth single problem, and then it is smoothed by proposed functions. Finally, the smoothed problem is solved using both of the proposed approaches. The presented approaches achieve an efficient and feasible solution in an appropriate time which has been evaluated by comparing to references and test problems.

Keywords: Linear bi-level programming problem, Linear tri -level programming problem, Karush-Kuhn-Tucker conditions, Genetic theorem, Line search method.

Cite this article: Hosseini, E., & Kamalabadi, I. N. (2014). Line Search and Genetic Approaches for Solving Linear Tri-level Programming Problem. *International Journal of Management, Accounting and Economics, 1* (4), 264-283.

[1] *Corresponding author's email: eghbal_math@yahoo.com*

Introduction

It has been proved that the BLP is NP- Hard problem even to seek for the locally optimal solutions (Bard, 1991; Vicente, et al., 1994). Nonetheless the BLPP is an applicable problem and practical tool to solve decision making problems. It is used in several areas such as transportation, finance and so on. Therefore finding the optimal solution has a special importance to researchers. Several algorithms have been presented for solving the BLP (Yibing, et al., 2007; Allende & G. Still,2012; Mathieu, et al., 1994; Wang, et al., 2008; Wend & U. P. Wen, 2000; Bard, 1998, Facchinei, et al., 1999). These algorithms are divided into the following classes: Transformation methods (Luce, et al., 2013; Dempe & Zemkoho, 2012), Fuzzy methods (Sakava et al., 1997; Sinha 2003; Pramanik & T.K. Ro 2009; Arora & Gupta 2007; Masatoshi & Takeshi.M 2012; Zhongping & Guangmin. 2008, Zheng et al., 2014), Global techniques (Nocedal & Wright, 2005; Khayyal, 1985; Mathieu, et al., 1994; Wang et al., 2008, Wan, et al., 2014, Xu, et al., 2014, Hosseini and Nakhai Kamalabadi, 2014), Primal–dual interior methods (Wend & Wen, 2000), Enumeration methods (Thoai et al., 2002), Meta heuristic approaches (Hejazi et al., 2002; Wang et al., 2008; Hu et al., 2010; Baran et al., 2010; Wan et al., 2012; Yan, et al., 2013; Kuen-Ming et al., 2007, Hosseini and Nakhai Kamalabadi., 2013, He, Li and Huang, 2014).However several algorithms have been proposed to BLPP, a few algorithms have been proposed to solve TLPP (Zhang et al., 2010).

The remainder of the paper is structured as follows: in Section 2, basic concepts of the linear BLPP and TLPP are introduced. We provide a smooth method to BLPP and TLPP in Section 3. The presented algorithm is proposed in Section 4. Computational results are presented for our approach in Section 5. Finally, the paper is finished in Section 6 by presenting the concluding remarks.

The linear bi-level and tri-level programming problems

In this section models of bi-level and tri-level programming problems are introduced.

BLPP is used frequently by problems with decentralized planning structure. It is defined as:

$$
\begin{aligned}
&\min_{x} F(x,y) = a^T x + b^T y \\
&\quad s.t\ \min_{y} g(x,y) = c^T x + d^T y \\
&\qquad\quad Ax + By \le r, \\
&\qquad\qquad\quad x, y \ge 0.
\end{aligned}
\tag{1}
$$

where $a, c \in R^{n_1}, b, d \in R^{n_2}, A \in R^{m\times n_1}, B \in R^{m\times n_2}, r \in R^m, x \in R^{n_1}, y \in R^{n_2}$ and $F(x,y)$ and $g(x,y)$ are the objective functions of the leader and the follower, respectively.

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

In general, BLPP is a non-convex optimization problem; therefore, there is no general algorithm to solve it. This problem can be non-convex even when all functions and constraints are bounded and continuous. Of course, the linear BLPP is convex and preserving this property is very important. A summary of important properties for convex problem are as follows, which $F: S \to R^n$ and S is a nonempty convex set in R^n:

(1) The convex function f is continuous on the interior of S.

(2) Every local optimal solution of F over a convex set $X \subseteq S$ is the unique global optimal solution.

(3) If $\nabla F(\bar{x}) = 0$, then $\bar{x}$ is the unique global optimal solution of F over S.

Because a tri-level decision reflects the principle features of multi-level programming problems, the algorithms developed for tri-level decisions can be easily extended to multi-level programming problems which the number of levels is more than three. Hence, just tri-level programming is studied in this paper.

In a TLPP, each decision entity at one level has its objective and its variables in part controlled by entities at other levels. To describe a TLPP, a basic model can be written as follows:

$$
\begin{aligned}
&\min_{x} F_1(x, y, z) = a_1 x + b_1 y + c_1 z \\
&\qquad A_1 x + B_1 y + C_1 z \le r_1, \\
&\quad \text{s.t} \min_{y} F_2(x, y, z) = a_2 x + b_2 y + c_2 z \\
&\qquad A_2 x + B_2 y + C_2 z \le r_2, \\
&\qquad \text{s.t} \min_{z} F_3(x, y, z) = a_3 x + b_3 y + c_3 z \\
&\qquad A_3 x + B_3 y + C_3 z \le r_3, \\
&\qquad\quad x, y, z \ge 0.
\end{aligned} \tag{2}
$$

Where $A_i \in R^{q\times k}, B_i \in R^{q\times l}, C_i \in R^{q\times p}, r_i \in R^q, x \in R^k, y \in R^l, z \in R^p, a_i \in R^k, b_i \in R^l, c_i \in R^p, i = 1,2,3$, and the variables x, y, z are called the top-level, middle-level, and bottom-level variables respectively, $F_1(x, y, z), F_2(x, y, z), F_3(x, y, z)$, the top-level, middle-level, and bottom-level objective functions, respectively. In this problem each level has individual control variables, but also takes account of other levels' variables in its optimization function.

To obtain an optimized solution to TLP problem based on the solution concept of bi-level programming, we first introduce some definitions and notation:

Definition 1

The feasible region of the TLP problem when i=1, 2, 3, is

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

$$S = \{(x, y, z) | A_i x + B_i y + C_i z \leq r_i\ , x, y, z \geq 0.\} \quad (3)$$

On the other hand, if x be fixed, the feasible region of the follower can be explained as

$$S = \{(y, z) | B_i y + C_i z \leq r_i - A_i x\ , y, z \geq 0\}. \quad (4)$$

Based on the above assumptions, the follower rational reaction set is

$$P(x) = \{(y, z) \in \operatorname{argming}(x, y, z), (y, z) \in S(x)\}. \quad (5)$$

Where the inducible region is as follows

$$IR = \{(x, y, z) \in S, (y, z) \in P(x)\}. \quad (6)$$

Finally, the tri-level programming problem can be written as

$$\min \{F(x, y, z) | (x, y, z) \in IR\}. \quad (7)$$

If there is a finite solution for the TLP problem, we define feasibility and optimality for the TLP problem as

$$S = \{(x, y, Z) | A_i x + B_i y + C_i z \leq r_i,\ \ x, y, z \geq 0\}. \quad (8)$$

Definition 2

Every point such as $(x, y, z)$is a feasible solution to tri-level problem if $(x, y, Z) \in IR$

Definition 3

Every point such as (x^*, y^*, z^*) is an optimal solution to the tri-level problem if

$$F(x^*. y^*, z^*\ ) \leq F(x, y, z)\ \ (x, y, z) \in IR. \quad (9)$$

Smooth method for TLPP

Using KKT conditions for both of last levels in problem (2), the following problem is constructed:

$$\min_{x} F_1(x, y, z) = a_1 x + b_1 y + c_1 z \quad (10)$$

$$\text{s.t}\quad A_1x + B_1y + C_1z - r_1 \leq 0,$$
$$A_2x + B_2y + C_2z - r_2 \leq 0,$$
$$A_3x + B_3y + C_3z - r_3 \leq 0,$$
$$\mu(A_3x + B_3y + C_3z - r_3) = 0,$$
$$\mu B_3 = -b_3,$$
$$\beta(A_2x + B_2y + C_2z - r_2) = 0,$$
$$\beta C_2 = -c_2,$$
$$x, y, z, \mu, \beta \geq 0.$$

Because problem (10) has a complementary constraint, it is not convex and it is not differentiable. In this paper we propose a smooth method for smoothing complementary constraints in problem (10). Using the following smooth method, problem (10) will be smoothed, and then we present two algorithms based on Taylor theorem and hybrid algorithm to solve it.

Theorem 3.1:

Let, $\phi: R^2 \to R$, $\phi(m,n) = 2m - n - \sqrt{4m^2 + n^2}$ or $\phi: R^3 \to R$, $\phi(m,n,\varepsilon) = 2m - n - \sqrt{4m^2 + n^2 + \varepsilon}$, where $m \geq 0$, $n \geq 0$, then $\phi(m,n) = 0 \Leftrightarrow mn = 0$, and $\phi(m,n,\varepsilon) = 0 \Leftrightarrow mn = \frac{\varepsilon}{4}, m \geq 0, n \geq 0$

Proof:

$$\phi(m,n) = 0 \Leftrightarrow 2m - n - \sqrt{4m^2 + n^2} = 0$$

$$\Leftrightarrow 2m - n = \sqrt{4m^2 + n^2} \Leftrightarrow (2m - n)^2 = 4m^2 + n^2$$

$$\Leftrightarrow 4m^2 + n^2 - 4mn = 4m^2 + n^2 \Leftrightarrow -4mn = 0 \Leftrightarrow mn = 0.$$

Also

$$\phi(m,n,\varepsilon) = 0 \Leftrightarrow 2m - n - \sqrt{4m^2 + n^2 + \varepsilon} = 0$$

$$\Leftrightarrow 2m - n = \sqrt{4m^2 + n^2 + \varepsilon} \Leftrightarrow (2m - n)^2 = 4m^2 + n^2 + \varepsilon$$

$$\Leftrightarrow 4m^2 + n^2 - 4mn = 4m^2 + n^2 + \varepsilon \Leftrightarrow -4mn = \varepsilon \Leftrightarrow mn = \frac{\varepsilon}{4}, m \geq 0, n \geq 0.$$

Using the proposed function $\phi(m,n,\varepsilon) = 2m - n - \sqrt{m^2 + n^2 - \varepsilon}$ in problem (10), we obtain the following problem: (11)

$$\min_{x} F_1(x, y, z) = a_1x + b_1y + c_1z$$

s.t

$$A_1x + B_1y + C_1z - r_1 \le 0,$$

$$A_2x + B_2y + C_2z - r_2 \le 0,$$

$$A_3x + B_3y + C_3z - r_3 \le 0,$$

$$2\mu_i - g_i(x, y) - \sqrt{4\mu_i^2 + g_i^2(x, y) - \varepsilon} = \frac{\varepsilon}{4}, i = 1,2, \dots, l,$$

$$\mu B_3 = -b_3,$$

$$2\beta_i - h_i(x, y) - \sqrt{4\beta_i^2 + h_i^2(x, y) - \varepsilon} = \frac{\varepsilon}{4}, i = 1,2, \dots, l,$$

$$\beta C_2 = -c_2,$$

$$x, y, z, \mu_i, \beta_i \ge 0.$$

Which in the first constraint $m = \mu_i \ge 0, n = -g_i(x, y) \ge 0,$ $g_i(x, y) = a^i x + b^i y + c^i z$ and a^i, b^i, c^i are i-th row of A, B, C respectively and in the second constraint $m = \beta_i \ge 0, n = -h_i(x, y) \ge 0,$ $h_i(x, y) = a^i x + b^i y + c^i z - r$ and a^i, b^i, c^i are i-th row of A, B, C.

Let:

G(x,y,μ)=

$$\begin{bmatrix} 2\mu_1 - g_1(x, y) - \sqrt{\mu_1^2 + g_1^2(x, y) - \varepsilon} \\ 2\mu_2 - g_2(x, y) - \sqrt{\mu_2^2 + g_2^2(x, y) - \varepsilon} \\ \vdots \\ 2\mu_l - g_l(x, y) - \sqrt{\mu_l^2 + g_l^2(x, y) - \varepsilon} \end{bmatrix} \qquad (12)$$

H(x,y, β)=

$$\begin{bmatrix} 2\beta_1 - h_1(x, y) - \sqrt{\mu_1^2 + h_1^2(x, y) - \varepsilon} \\ 2\beta_2 - h_2(x, y) - \sqrt{\mu_2^2 + h_2^2(x, y) - \varepsilon} \\ \vdots \\ 2\beta_l - h_l(x, y) - \sqrt{\mu_l^2 + h_l^2(x, y) - \varepsilon} \end{bmatrix} \qquad (13)$$

ISSN 2383-2126 (Online)

$$H'(x, y, \mu) = H(x, y, \mu) - \frac{\varepsilon}{4} \quad ,$$
$$G'(x, y, \mu) = G(x, y, \beta) - \frac{\varepsilon}{4} \qquad (14)$$

Problem (11) can be written as follows:

$$\min_{x} F_1(x, y, z) = a_1 x + b_1 y + c_1 z$$

$$\text{s.t}$$

$$\begin{aligned}
&A_1 x + B_1 y + C_1 z - r_1 \leq 0,\\
&A_2 x + B_2 y + C_2 z - r_2 \leq 0,\\
&A_3 x + B_3 y + C_3 z - r_3 \leq 0,\\
&G'(t) = 0,\\
&H'(t) = 0,\\
&\mu B = -b_3,\\
&\beta C = -c_2,\\
&\qquad x, y, z, \mu, \beta \geq 0.
\end{aligned} \qquad (15)$$

Where $t = (x, y, \mu) \in R^{k+2l}$

Because problem (10) equal to (15), we use the following method for solving problem (15).

Genetic algorithm for TLPP

In this section, basic and general concepts related to genetic proposed algorithm are discussed. Genetic algorithms are global methods that are used for global searches. As the previous researchers indicate (Luce, Saïd & Raïd, 2013), Masatoshi & Takeshi (2012) the basic characteristics of these algorithms consist of:

1. Initial population of solution is produced randomly. Some of the genetic algorithms use other Meta heuristic method to produce the initial population.
2. Genetic algorithms use a lot of feasible solutions. Therefore they usually avoid local optimal solutions.
3. Genetic algorithms used to solve very large problems with many variables.
4. These algorithms are simple and do not need extra conditions such as continuity and differentiability of objective functions.
5. Genetic algorithms usually gain several optimal solutions instead unique optimal solution. This property is useful for multi objective function and multi- level programming.

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

In the proposed genetic algorithm, each feasible solution of BLPP usually is transformed by string of characters from the binary alphabet that is called chromosome. The genetic algorithm works as follows:

Initial generation, that is generated randomly, is divided in overall the feasible space similarly. Then chromosomes are composed together to construct new generation. This process continues till to get appropriate optimal solution. The general genetic algorithm process as follows:

Algorithm 1 : GA to solve BLPP

1 : $t = 0$
2 : initialize $P(t)$
3 : evaluate $P(t)$
4 : While not *terminate* do
5 : $P'(t) =$ recombine $P(t)$
6 : $P''(t) =$ mutate $P'(t)$
7 : evaluate $P''(t)$
8 : $P(t+1) =$ select $(P''(t) \cup Q)$
9 : $t = t + 1$
10 : End of While
11 : End.

Where *P(t)* is a population of chromosomes in *t-th* generation and *Q* is a set of chromosomes in the current generation which are selected.

In the suggested method, every chromosome is demonstrated by a string. This string consists of

$k + l + p$, binary components. Also these chromosomes are applied in problem (15) that it is created by using Karush -Kuhn –Tucker (KKT) conditions and proposed smoothed method for TLPP. Using slack variables, such as w, v, u, problem (15) is prepared for using genetic algorithm:

$$\min_{x} F_1(x, y, z) = a_1 x + b_1 y + c_1 z$$

$$\text{s.t} \qquad (16)$$

$$A_1 x + B_1 y + C_1 z - r_1 + w = 0,$$

$$A_2 x + B_2 y + C_2 z - r_2 + v = 0,$$

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

$$A_3x + B_3y + C_3z - r_3 + u = 0,$$

$$G'(t) = 0,$$

$$H'(t) = 0,$$

$$\mu B = -b_3,$$

$$\beta C = -c_2,$$

$$tw = 0, \qquad vy = 0, \quad uz = 0,$$

$$t, w, v, u, x, y, z, \mu, \beta \geq 0.$$

Now the chromosomes are applied according the following rules [15]:

If the *i-th* component of the chromosome is equal to zero, then $t_i = 0, w_i \geq 0$ Else $t_i \geq 0, w_i = 0$.

If the *j-th* component of the chromosome is equal to zero, then $v_j = 0, y_j \geq 0$ Else $v_j \geq 0, y_j = 0$.

If the *h-th* component of the chromosome is equal to zero, then $u_h = 0, z_h \geq 0$ Else $u_h \geq 0, z_h = 0$.

Steps of our algorithm

In this section, the algorithm steps are proposed.

Step 1: Generating the initial population.

The initial population includes solutions in the feasible region that are called achievable chromosomes. These chromosomes are generated by solving the following problem:

$$\min_{z} c_3 z$$

$$A_3x + B_3y + C_3z \leq r_3,$$

$$z \geq 0.$$

Where, r_3 is a random vector by changing it, the optimal solution changes too.

Step 2: Keeping the present best chromosome in an array

The best chromosome is kept in the array at the each iteration. This process continues till the algorithm is finished, then the best chromosome is found in the array as the optimal solution.

ISSN 2383-2126 (Online)

Step 3: Crossover operation

Crossover is a major operation to compose a new generation. In this stage two chromosomes are selected randomly and they are combined to generate a new chromosome. In the new generation components are created by the following rules:

1. The i-th component of the first child is replaced by the sum of the i-th components of parents (i=1,2,…,k). The operation sum is defined as follows:

$$0+1=1 \qquad 0+1=1 \qquad 0+0=0 \qquad 1+1=0$$

The other components are remained the same as the first parent.

2. The (k+i)-th component of the second child is replaced by the sum of the (k+i)-th components of parents (i=1,2,…,l+p). The operation sum is defined as above. The other components are remained the same as the second parent.

For example, by applying the present method to the following parents, and k= 5, l=4, p=3, we generate the following children:

Parents:	Children:
10110 1001 101	01100 1001 101
11010 0111 001	11010 1110 100

Step 4: Mutation

The main goal of mutation in GA is to avoid trapping in local optimal solutions. In this algorithm each chosen gene of every chromosome, mutates as follows:

If the value of the chosen gene be 0, it will be changed to 1 and if the value of the chosen gene be 1, it will be changed to 0.

Step 5: Selection

The chromosomes of the current population are arranged in descending order of fitness values. Then we select a new population similar to the size of the first generation. If the number of the generations is sufficient we go to the next step, otherwise the algorithm is continued by the step3.

Step 6: Termination

The algorithm is terminated after a maximum generation number. The best produced solution that has been recorded in the algorithm is reported as the best solution to BLPP by proposed GA algorithm.

Hybrid algorithm (HA)

Penalty functions transform a constrained problem into a single unconstrained problem or into a sequence of unconstrained problems. The constraints are appended into the objective function via a penalty parameter in a way that penalizes any violation of the constraints. In general, a suitable function must incur a positive penalty for infeasible points and no penalty for feasible points. Also, the penalty function method is a common approach to solve the bi-level programming problems. In this kind of approach, the lower level problem is appended to the upper level objective function with a penalty. We use a penalty function to convert problem (15) to an unconstraint problem.

Consider problem (15); we append all constraints to the first level objective function with a penalty for each constraint. Then, we obtain the following penalized problem.

$$\min_{t} \mathrm{F}(t) + \alpha_1(\beta\mathrm{C} + c_2)^2 + \alpha_2(\mu B + b_3)^2 + \alpha_3\left(G'(\mathrm{t})\right)^2 + \alpha_4\left(H'(\mathrm{t})\right)^2 + \sum_i \alpha_i(A_i\mathrm{x} + B_i\mathrm{y} + C_i\mathrm{z} - r_i)^2 \qquad (17)$$

Now we solve problem (15) using our line search method. The line search method is proposed as follows:

Given a vector x, a suitable direction d is first determined, and then f is minimized from x in the direction d. Our method searches along the directions $(d_1, d_2, \dots, d_{n-1})$ where $d_j, j = 1,2, \dots, n-1$ is a vector of zeros except at the jth position which is 1 and $d_n = \left(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, , \dots, \frac{1}{\sqrt{n}}\right)$.

Clearly, all directions have a norm equal to 1 and they are linearly independent search directions. In fact, the proposed line search method uses the following directions as the search directions:

$$d_1 = (1,0, \dots ,0), d_2 = (0,1, \dots ,0), \dots , d_{n-1} = (0, \dots ,1,0), d_n = \left(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, , \dots, \frac{1}{\sqrt{n}}\right) \qquad (18)$$

Therefore, along the search direction $d_j, j = 1,2, \dots, n-1$, the variable x_j is changed while all other variables are kept fixed. We summarize below the proposed line search method for minimizing a function of several variables. Then, we show that, if the function is differentiable then the proposed method converges to a stationary point.

Step 1: Initial step

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

Choose a scalar $\varepsilon > 0$ to be used for terminating the algorithm, and let $d_1, d_2, \ldots, d_{n-1}$ be the coordinate directions and d_n be a vector of $\frac{1}{\sqrt{n}}$. Choose an initial point x_1 let $x_1 = y_1. k = j = 1$, and go to the next step.

Step 2: Main step

Let μ_j be an optimal solution to the problem to minimize $(y_j + \mu d_j)$, and let $y_{j+1} = y_j + \mu_j d_j$

If $j < n$ replace j by $j + 1$, and repeat step1. Otherwise, if $j = n$, go to the next step.

Step 3: Termination

Let $x_{k+1} = y_{n+1}$ if $\|x_{k+1} - x_k\| < \varepsilon$ then stop, otherwise, let $y_1 = x_{k+1}$ and $j = 1$, replace k by $k + 1$, and repeat step 2.

We now propose a theorem which establishes the convergence of algorithms for solving a problem of the form: minimize $f(x)$ subject to $x \in R^n$. We show that an algorithm that generates n linearly independent search directions, and obtains a new point by sequentially minimizing f along these directions, converges to a stationary point. The theorem also establishes the convergence of algorithms using linearly independent and orthogonal search directions.

We now show that two problems (15) and (17) have the same optimal solution according to the following theorem.

Theorem 5.1:

Consider the following problem:

$$\min_x f(x)$$

$$s.t\ g_i(x) \leq 0,\ \text{i=1,2,…,m,} \qquad (19)$$

$$h_j(x) = 0,\ \text{j=1,2,…,l,}$$

where $f, g_1, \ldots, g_m, h_1, \ldots, h_l$ are continuous functions on R^n and X is a nonempty set in R^n. Suppose that the problem

has a feasible solution, and α is a continuous function as follows:

$$\alpha(\mathrm{x}) = \sum_{i=1}^{m} \emptyset[g_i(x)] + \sum_{i=1}^{l} \emptyset[h_i(x)] \qquad (20)$$

where

$$\emptyset(y) = 0 \text{ if } \mathrm{y} \leq 0,\ \emptyset(y) > \qquad (21)$$

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

$0 \; if \;\; y > 0.$

$\emptyset(y) = 0$ if y = 0, $\emptyset(y) > 0 \; if \;\; y \neq 0.$ (22)

Then,

$$\inf\{f(x): g(x) \leq 0, \;\; h(x) = 0, x \in X\} = \inf\{f(x) + \mu\alpha(x): x \in X\} \qquad (23)$$

Where μ is a large positive constant (μ → ∞).

Proof:

This theorem has been proven by (Nocedal, J, & S.J. Wright. (2005)).

Computational results

To illustrate both algorithms, we consider the following examples.

Example 1 (Zhang , G, J. Lu , J. Montero , & Y. Zeng , Model. (2010)**):**

Consider the following linear tri-level programming problem:

$$\min_{x} x - 4y + 2z$$

$$\text{s.t}$$

$$-x - y \leq -3,$$

$$-3x + 2y - z \geq -10,$$

$$\min_{y} x + y - z$$

$$\text{s.t}$$

$$-2x + y - 2z \leq -1,$$

$$2x + y + 4z \leq 14,$$

$$\min_{y} x - 2y - 2z$$

$$\text{s.t}$$

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

www.ijmae.com

$$2x - y - z \leq 2,$$

$$x, y, z \geq 0.$$

Using KKT conditions, the following problem is obtained:

$$\min_{x} x - 4y + 2z$$

$$\text{s.t}$$

$$-x - y \leq -3,$$

$$3x - 2y + z \leq 10,$$

$$-2x + y - 2z \leq -1,$$

$$2x + y + 4z \leq 14,$$

$$\beta_1(-2x + y - 2z + 1) = 0,$$

$$\beta_2(2x + y + 4z - 14) = 0,$$

$$\beta_1 + \beta_2 = 1,$$

$$2x - y - z \leq 2,$$

$$\mu(2x - y - z - 2) = 0,$$

$$\mu(-1) = -2,$$

$$x, y, z, \beta_1, \beta_2, \mu \geq 0.$$

By the proposed function, the above problem becomes:

$$\min_{x} x - 4y + 2z$$

$$\text{s.t}$$

$$-x - y \leq -3,$$

$$3x - 2y + z \leq 10,$$

$$2\beta_1 - (-2x + y - 2z + 1) - \sqrt{\beta_1{}^2 + (-2x + y - 2z + 1)^2 + \varepsilon} = 0,$$

International Journal of Management, Accounting and Economics
Vol. 1, No. 4, November, 2014
ISSN 2383-2126 (Online)

$$2\beta_2 - (2x + y + 4z - 14) - \sqrt{\beta_2{}^2 + (2x + y + 4z - 14)^2 + \varepsilon} = 0,$$

$$2\mu - (2x - y - z - 2) - \sqrt{\mu^2 + (2x - y - z - 2)^2 + \varepsilon} = 0,$$

$$\beta_1 + \beta_2 = 1,$$

$$\mu\,(-1) = -2,$$

$$x, y, z, \beta_1, \beta_2, \mu \geq 0.$$

Optimal solution presented according to Table 1. Behavior of the variables in Example 1 has been show in figure 1.

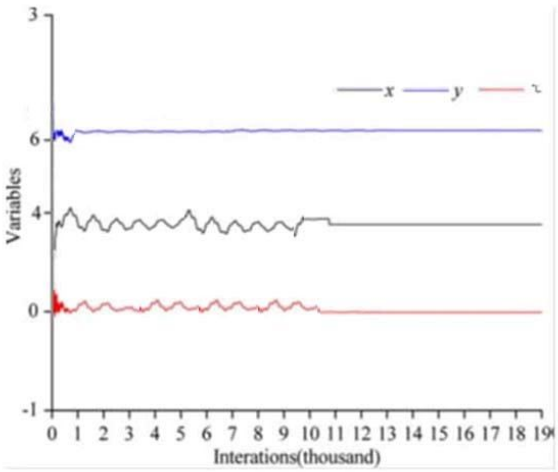

Figure 1- Behavior of the variables in Example 1

Table 1- Comparison of optimal solutions by genetic algorithm – Example 1.

Optimal Solution	Best solution by our method	Best solution according to reference
(x^*, y^*, z^*)	(4,6,0)	(4,6,0)
F_1(x, y, z)	-20	-20
F_2(x, y, z)	10	10
F_3(x, y, z)	-8	-8

Example 2 (Zhang , G, J. Lu , J. Montero , & Y. Zeng , Model. (2010).**):**

Consider the following linear tri-level programming problem.

ISSN 2383-2126 (Online)

$$\min_{x} x + 4y + 2z$$
$$\text{s.t}$$
$$x - 3y + 9z \leq 30,$$
$$-3x + 5y - z \leq -100,$$
$$\min_{y} -x + 7y - z$$
$$\text{s.t}$$
$$3x + 5y - z \leq 160,$$
$$\min_{y} 7x + y + 21z$$
$$\text{s.t}$$
$$3x - 4y - 2z \leq 212,$$
$$x, y, z \geq 0.$$

Optimal solution for this example is presented according to Table 2. Behavior of the variables has been show in figure 2.

Table 2- Comparison of optimal solutions by Taylor algorithm – Example 2.

Optimal Solution	Best solution by our method	Best solution according to reference
(x^*, y^*, z^*)	(10,28.33,11.66)	(10,28.33,11.66)
$F_1(x, y, z)$	146.66	146.66
$F_2(x, y, z)$	176.6	176.6
$F_3(x, y, z)$	343.3	343.3

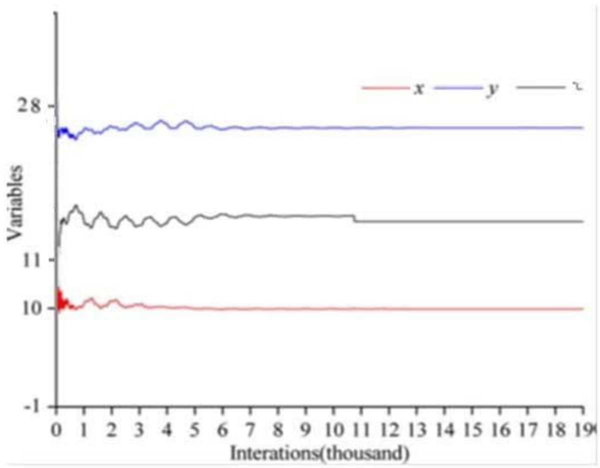

Figure 2 - Behavior of the variables in Example 2

Example 1 (solving by hybrid algorithm):

Consider the following linear tri-level programming problem:

$$\min_{x} x - 4y + 2z$$
$$\text{s.t}$$
$$-x - y \leq -3,$$
$$-3x + 2y - z \geq -10,$$
$$\min_{y} x + y - z$$
$$\text{s.t}$$
$$-2x + y - 2z \leq -1,$$
$$2x + y + 4z \leq 14,$$
$$\min_{y} x - 2y - 2z$$
$$\text{s.t}$$
$$2x - y - z \leq 2,$$
$$x, y, z \geq 0.$$

Using hybrid algorithm the problem is solved. Optimal solution for this example by hybrid algorithm is presented according to Table 3.

Table 3- Comparison of optimal solutions by hybrid algorithm – Example 1.

Optimal Solution	Best solution by hybrid algorithm	Best solution according to reference
(x^*, y^*, z^*)	(4.3,6.2,0.1)	(4,6,0)
$F_1(x, y, z)$	-20.3	-20
$F_2(x, y, z)$	10.4	10
$F_3(x, y, z)$	-8.3	-8

Example 2 is solved by hybrid algorithm and computational results are proposed in Table 4.

Table 4- Comparison of optimal solutions by hybrid algorithm – Example 2.

Optimal Solution	Best solution by hybrid algorithm	Best solution according to reference
(x^*, y^*, z^*)	(10.1,28.4,11.6)	(10,28.33,11.66)
$F_1(x, y, z)$	147.16	146.66
$F_2(x, y, z)$	176.93	176.6
$F_3(x, y, z)$	345.33	343.3

References

Allende, G, & B. G. Still. (2012). Solving bi-level programs with the KKT-approach, Springer and Mathematical Programming Society 1 31:37 – 48.

Arora, S.R, & R. Gupta. (2007). Interactive fuzzy goal programming approach for bi-level programming problem, European Journal of Operational Research 176 1151–1166.

Bard, J.F. (1991). Some properties of the bi-level linear programming, Journal of Optimization Theory and Applications 68 371–378.

Bard, J.F. (1998). Practical bi-level optimization: Algorithms and applications, Kluwer Academic Publishers, Dordrecht.

Dempe, S, & A.B. Zemkoho.(2012). On the Karush–Kuhn–Tucker reformulation of the bi-level optimization problem, Nonlinear Analy sis 75 1202–1218.

Facchinei, F, H. Jiang, & L. Qi. (1999). A smoothing method for mathematical programming with equilibrium constraints, Mathematical Programming 85 107-134.

He, X, & C. Li, T. Huang, C. Li. (2014).Neural network for solving convex quadratic bilevel programming problems, Neural Networks, Volume 51, March, Pages 17-25.

Hosseini, E, & I.Nakhai Kamalabadi. (2013).A Genetic Approach for Solving Bi-Level Programming Problems, Advanced Modeling and Optimization, Volume 15, Number 3.

Hosseini, E , & I.Nakhai Kamalabadi. (2013).Solving Linear-Quadratic Bi-Level Programming and Linear-Fractional Bi-Level Programming Problems Using Genetic Based Algorithm, Applied Mathematics and Computational Intellegenc, Volume 2.

Hosseini, E & I.Nakhai Kamalabadi. (2014). Taylor Approach for Solving Non-Linear Bi-level Programming Problem ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No.11 , September.

Hejazi, S.R, A. Memariani, & G. Jahanshahloo. (2002). Linear bi-level programming solution by genetic algorithm, Computers & Operations Research 29 1913–1925.

Hu, T. X, Guo, X. Fu, & Y. Lv, (2010). A neural network approach for solving linear bi-level programming problem, Knowledge-Based Systems 23 239–242.

Khayyal, A.AL. (1985). Minimizing a Quasi-concave Function Over a Convex Set: A Case Solvable by Lagrangian Duality, proceedings, I.E.E.E. International Conference on Systems, Man,and Cybemeties, Tucson AZ 661-663.

Kuen-Ming, L, Ue-Pyng.W, & Hsu-Shih.S. (2007). A hybrid neural network approach to bi-level programming problems, Applied Mathematics Letters 20 880–884

Luce, B, Saïd.H, & Raïd.M. (2013).One-level reformulation of the bi-level Knapsack problem using dynamic programming, Discrete Optimization 10 1–10.

Masatoshi, S, & Takeshi.M. (2012).Stackelberg solutions for random fuzzy two-level linear programming through possibility-based probability model, Expert Systems with Applications 39 10898–10903.

Mathieu, R, L. Pittard, & G. Anandalingam. (1994). Genetic algorithm based approach to bi-level Linear Programming, Operations Research 28 1–21.

Nocedal, J, & S.J. Wright. (2005). Numerical Optimization, Springer-Verlag, , New York.

Pramanik, S, & T.K. Ro. (2009). Fuzzy goal programming approach to multilevel programming problems, European Journal of Operational Research 194 368–376.

Sakava, M, I. Nishizaki, & Y. Uemura. (1997). Interactive fuzzy programming for multilevel linear programming problem, Computers & Mathematics with Applications 36 71–86.

Sinha, S. (2003). Fuzzy programming approach to multi-level programming problems, Fuzzy Sets And Systems 136 189–202.

Thoai, N. V, Y. Yamamoto, & A. Yoshise. (2002). Global optimization method for solving mathematical programs with linear complementary constraints, Institute of Policy and Planning Sciences, University of Tsukuba, Japan 978.

Vicente, L, G. Savard, & J. Judice. (1994). Descent approaches for quadratic bi-level programming, Journal of Optimization Theory and Applications 81 379–399.

Wend, W. T, & U. P. Wen. (2000). A primal-dual interior point algorithm for solving bi-level programming problems, Asia-Pacific J. of Operational Research, 17.

Wang, G. Z, Wan, X. Wang, & Y.Lv. (2008).Genetic algorithm based on simplex method for solving Linear-quadratic bi-level programming problem, Computers and Mathematics with Applications 56 2550–2555.

Wan, Z. G, Wang, & B. Sun. (2012). A hybrid intelligent algorithm by combining particle Swarm optimization with chaos searching technique for solving nonlinear bi-level programming Problems, Swarm and Evolutionary Computation.

Wan, Z, L. Mao, & G. Wang. (2014). Estimation of distribution algorithm for a class of nonlinear bilevel programming problems, Information Sciences, Volume 256, 20 January, Pages 184-196.

Xu, P, & L. Wang. (2014). An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions, Computers & Operations Research, Volume 41, January, Pages 309-318.

Yan, J, Xuyong.L, Chongchao.H, & Xianing.W. (2013). Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bi-level programming problem, Applied Mathematics and Computation 219 4332–4339.

Yibing, Lv, Hu. Tiesong, & Wang. Guangmin. (2007). A penalty function method Based on Kuhn–Tucker condition for solving linear bilevel programming, Applied Mathematics and Computation 1 88 808–813.

Zhang , G, J. Lu , J. Montero , & Y. Zeng , Model. (2010). solution concept, and Kth-best algorithm for linear tri-level, programming Information Sciences 180 481–492

Zhongping, W, & Guangmin.W. (2008).An Interactive Fuzzy Decision Making Method for a Class of Bi-level Programming, Fifth International Conference on Fuzzy Systems and Knowledge Discovery.

Zheng, Y, J. Liu, & Z. Wan. (2014). Interactive fuzzy decision making method for solving bi-level programming problem, Applied Mathematical Modelling, Volume 38, Issue 13, 1 July, Pages 3136-3141.