App. Math. and Comp. Intel., Vol. 2 (2) (2013) 169–182

http://amci.unimap.edu.my

Solving linear-quadratic bi-level programming and linear-fractional bi-level programming problems using genetic algorithm

Eghbal Hosseini[a,*], Isa Nakhai Kamalabadi[b]

[a] *Department of Mathematics , Faculty of science, University of Payamenur, Tehran, Iran*

[b] *Department of Industrial Engineering , Faculty of engineering, University of Kurdistan, Sanandaj, Iran*

Received: 1 Jul 2013; Accepted: 5 Aug 2013

Abstract: *The bi-level programming problem (BLPP) is a suitable method for solving the real and complex problems in applicable areas. There are several forms of the BLPP as an NP-hard problem. The linear-quadratic bi-level programming (LQBP) and the linear-fractional bi-level programming (LFBP) problems are two important forms of the BLPP. In this article, we show an effective method based on genetic algorithm (GA) for solving such problems. To obtain efficient upper bounds and lower bounds we use the Karush-Kuhn-Tucker (KKT) conditions for transforming the LQBP and the LFBP into single level problems. Thus by using the proposed GA, the single problems are solved. The proposed approach achieves efficient and feasible solutions and they are evaluated by comparing with references and test problems.*

Keywords: *Linear-fractional bi-level programming problem, Linear-quadratic bi-level programming problem, genetic algorithm.*

PACS: *02.60.Pn*

1. Introduction

The bi-level programming problem (BLPP) is a nested optimization problem, which has two levels in hierarchy. The first level is called leader and the second level is called follower which they have their own objective functions and constraints. It has been proved that the BLPP is NP- Hard problem even to seek for the locally optimal solutions [1, 2]. Nonetheless the BLPP is an applicable problem and practical tool for solving decision making problems. The BLPP is used in several areas such as economic, traffic, finance and so on. For example in congestion pricing problem which provides an optimal price for vehicles entering the bridges or specified areas a BLPP model is the best known model which in first level the income of the leader that in this case usually the municipal

* Corresponding Author: eghbal_math@yahoo.com. (E. Hosseini)

is maximized whenever in the second level the users or drivers are trying to minimize their route from origin to the destination.

Several algorithms have been presented for solving the BLPP [3, 4, 11-13]. These algorithms are divided into the following classes: Transformation methods [3, 4, 22, 23, 24], Fuzzy methods [5, 6, 7, 8, 25, 26], Global techniques [9, 10, 11, 12, 29, 30], Primal–dual interior methods [13], Enumeration methods [14], Metaheuristic approaches [15, 16, 17, 18, 19, 27-28].

In this paper, we consider two forms of the BLPP: the linear-quadratic bi-level programming (LQBP) which the objective function of the upper level is linear and the objective function of the lower level is quadratic and the linear-fractional bi-level programming (LFBP) which the objective function of the upper level is fractional and the objective function of the lower level is linear. It will be presented a procedure based on genetic algorithm to solve these two problems. In the remaining of pages, literature review is proposed in Section 2. In Section 3, basic concepts of LQBP and LFBP are proposed. We provide the GA for solving LQBP and LFBP in Section 4. Section 5 describes the steps of our algorithm. Computational results are proposed in Section 6 and finally, the paper is finished in section 7.

2 Literature review

R. Mathieu [11] has proposed a global method to solve BLPP by using genetic algorithm. This algorithm primarily produces feasible solutions which they eligible for optimized solutions to the follower objective function then in the next step algorithm provides the optimal solution for the leader objective function. W.T. Weng [13] proposed a primal-dual method to solve BLPP. S.R. Hejazi et al [15] presented a very efficient method based on genetic algorithm. This approach can solve the BLPP with several sizes. H. I. Calvete [21] solved the BLPP problem using a penalty function algorithm. Lv et al used KKT conditions to convert the BLPP into a single level problem. Then they append the complementary conditions to the high level objective function. Finally by decomposing the linear bi-level programming to a number of simpler and smaller linear programming problems, they solved the BLPP problem [3]. Wang and Wan constructed a genetic method based on the simplex algorithm. This method, which is proposed to solve the linear quadratic bi-level programming problem, first transforms the problem into a single level problem by using KKT conditions then solves the single level problem which can be simplified to a linear problem [16]. Baran et al [18] proposed a global method based on genetic algorithm to solve fuzzy quadratic bi-level programming problem. Hu and Guo [17] also proposed a new neural network for solving linear bi-level problem. This method not only has been proved to be stable but also it can generate the optimal solution. Allende and Still surveyed the advantages and disadvantages of the Karush-Kuhn-Tucker conditions [4]. Wan et al [19], proposed a hybrid intelligent algorithm by combining the particle swarm optimization with chaos searching technique (CST) for solving nonlinear bi-level programming problems. In this paper, the algorithm is initialized by a set of random particles. Then, an optimization problem is solved by CST. B. Luce [23] introduced a new reformulation of the bi-level knapsack problem (BKP). In this paper an algorithm was proposed based dynamic programming for solving the BKP. J. Yan [27], using Karush-Kuhn-Tucker (KKT) condition to the lower level problem, transformed the non-linear bi-level programming (NBLP) into a regular nonlinear programming with complementary constraints. Then by particle swarm optimization (PSO) approach solved the smoothed nonlinear programming.

3 Bi-level Programming Problem: Concepts & Properties

In this research we propose only two special classes of bi-level programming: linear-quadratic bi-level programming (LQBP) and Linear-fractional bi-level programming (LFBP). The LQBP is formulated as follows [16]:

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

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), g (x, y) are the objective functions of the leader and the follower, respectively. Also $Q \in R^{n_1+n_2} \times R^{n_1+n_2}$ is symmetric positive semi –definite matrix.

Suppose that

$$
Q = \begin{bmatrix} Q_2 & Q_1^T \\ Q_1 & Q_0 \end{bmatrix}
$$

Which $Q_0 \in R^{n_2 \times n_2}$, $Q_1 \in R^{n_2 \times n_1}$, $Q_2 \in R^{n_1 \times n_1}$.

Then the follower problem of the LQBP is

$$
\begin{aligned}
&\max_{y} \quad g(x,y) = d^T y + 2Q_1 xy + y^T Q_0 y \qquad (2) \\
&\qquad s.t \quad By \le r - Ax, \\
&\qquad\qquad y \ge 0.
\end{aligned}
$$

The LFBP problem is formulated as follows [21]:

$$
\begin{aligned}
&\max_{x} \quad f(x,y) = \frac{a_1 + a_2^T x + a_3^T y}{b_1 + b_2^T x + b_3^T y} \\
&\quad s.t \ \max_{y} \quad g(x,y) = c^T x + d^T y \qquad (3) \\
&\qquad\quad s.t \quad Ax + By \le r, \\
&\qquad\qquad\quad x, y \ge 0.
\end{aligned}
$$

Which a_1 , $b_1 \in R$, a_2 , $b_2, c \in R^{n_1}$, a_3, b_3 , $d \in R^{n_2}$, $A \in R^{m \times n_1}$, $B \in R^{m \times n_2}$, $r \in R^m$, $x \in R^{n_1}$, $y \in R^{n_2}$.

The feasible region of the LQBP and LFBP problems is

$$S = \{(x, y) \mid Ax + By \le r \ \ , \ \ x, y \ge 0\}. \qquad (4)$$

Which, this region is nonempty.

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

$$S(x) = \{y \mid By \le r - Ax \ \ , \ \ x, y \ge 0\}. \qquad (5)$$

Based on the above assumptions the follower rational reaction set can be shown as

$$P(x) = \{y \mid y \in \arg\max[g(x, y) \mid y \in S(x)\}. \qquad (6)$$

Where the inducible region is as follows

$$IR = \{(x, y) \in S, y \in P(x)\}. \qquad (7)$$

Finally the bi-level programming problem can be written as

$$\max\{f(x, y) \mid (x, y) \in IR\}. \qquad (8)$$

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

$$S = \{(x, y) \mid Ax + By \le r, x, y \ge 0\}. \qquad (9)$$

Definition 1:

(x, y), is a feasible solution to bi-level problem if $(x, y) \in IR.$

Definition 2:

(x^*, y^*) is an optimal solution to the bi-level problem if

$$f(x^*, y^*) \leq f(x, y) \quad \forall (x, y) \in IR. \tag{10}$$

4 The Proposed Genetic Algorithm (GA-LQBP/GA-LFBP)

In this section, basic and general concepts of GA-LQBP/GA-LFBP are discussed. Genetic algorithms are global methods which are used for global searches. The basic characteristics of these algorithms, as the previous researchers indicate [11, 15-16], consist of:

1. Initial population of feasible solutions is produced randomly. Some of the genetic algorithms use other Metaheuristic methods to produce the initial population.
2. Genetic algorithms use many of feasible solutions therefore they usually avoid local optimal solutions.
3. Genetic algorithms can solve large problems with many variables.
4. These algorithms are simple and they 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.

The parameters which are used, as follows:

U and v are used by KKT optimality conditions.

W is a positive slack variable which convert inequality constraint into equality constraint.

P (t) is population of chromosomes in $t.th$ generation.

M is number of constraints in the lower level problem in the BLPP.

N is number of variables in the lower level problem in the BLPP, and ε is given positive small number.

In the GA-LQBP/GA-LFBP, each feasible solution of BLPP usually is transformed by string of characters from the binary alphabet that is called chromosome. This 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

$t = 0$
initialize $P(t)$
evaluate $P(t)$
While not *terminate* do
$P'(t) = \text{recombine } P(t)$
$P''(t) = \text{mutate } P'(t)$ (11)
evaluate $P''(t)$
$P(t+1) = \text{select } (P''(t) \cup Q)$
$t = t + 1$
End of While
End.

Figure 1: the general genetic algorithm process

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 $m + n_2$ binary components. Also these chromosomes are applied in the

The following problems that they are created by using Karush-Kuhn-Tucker (KKT) conditions for LQBP and LFBP respectively:

$$\begin{aligned} \max\ & a^T x + b^T y \\ s.t\ & Ax + By + w = r \\ & 2Q_1 x + 2Q_0 y - Bu + v = -d \\ & uw = 0\ ,\ vy = 0, \\ & x, y, u, v, w \geq 0. \end{aligned} \tag{12}$$

$$\begin{aligned} \max_x\ & f(x, y) = \frac{a_1 + a_2^T x + a_3^T y}{b_1 + b_2^T x + b_3^T y} \\ s.t\ & Ax + By + w = r, \\ & B^T u - v = d, \\ & uw = 0,\ vy = 0, \\ & x, y, u, v, w \geq 0 \end{aligned} \tag{13}$$

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

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

Theorem 1:

(x^*, y^*), is the optimal solution to the problem (1) if and only if there exists u^* , w^* , v^* , such that $(x^*, y^*, u^*, w^*, v^*)$ is the solution of the problem (12).

Proof :

Theproof of this theorem was given by [16].

Theorem 2:

(x^*, y^*), is the optimal solution to the problem (3) if and only if there exists u^* , w^* , v^*, such that $(x^*, y^*, u^*, w^*, v^*)$ is the solution of the problem (13).

Proof :

Theproof of the theorem was shown by reference [12].

5 The steps of algorithm GA-LQBP/GA-LFBP

The algorithm for solving the LQPP/LFPP problems by genetic algorithm is proposed as follow:

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 to the LQBP and LFBP respectively:

$$\begin{aligned} \max \quad & (d + 2Q_1 x)^T y + y^T Q_0 y \\ s.t \quad & Ax + By \leq r, \\ & x, y \geq 0. \end{aligned} \qquad (14)$$

$$\begin{aligned} \max_y \quad & d^T y \\ s.t \quad & Ax + By \leq r, \\ & x, y \geq 0. \end{aligned} \qquad (15)$$

, which *r* is a random vector that changes the optimal solution.

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.

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 $i = (1,2,...,m)$. The operation sum is defined as follows:

1+0=1	0+1=1	0+0=0	1+1=0

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

2. The $(m+i)-th$ component of the second child is replaced by the sum of the $(m+i)-th$ components of parents $(i = 1,2,...,n_2)$. 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 m= 5, $n_2 = 4$, we generate the following children:

Parents:	Children:
10110 1001	01100 1001
11010 0111	11010 1110

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 or whenever:

$$\left| x_{n+1} - x_n \right| < \varepsilon$$

, which ε is a small positive number and x_n, x_{n+1} are the best solutions at the n and n+1 iteration. The best produced solution that has been recorded in the algorithm is reported as the best solution to BLPP by proposed GA algorithm.

6. Computational results

To illustrate the feasibility and efficiency of the GA-LQBP/GA-LFBP two following examples are solved. The first example is LQBP and the second example is LFBP.

Example 1:

Consider the following linear quadratic bi-level programming problem [16].

$$\begin{aligned}
&\max_x \ x_1 + x_2 + 3y_1 - y_2 \\
&\quad s.t \ \max_y \quad 5y_1 + 8y_2 + (x_1, x_2, y_1, y_2) \begin{pmatrix} 1 & 3 & 2 & 0 \\ 3 & 1 & 4 & -2 \\ 2 & 4 & -2 & 1 \\ 0 & -2 & 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \\
&\qquad\quad s.t \quad x_1 + x_2 + y_1 + y_2 \le 12, \\
&\qquad\qquad -x_1 + x_2 \le 2, \\
&\qquad\qquad 3x_1 - 4y_2 \le 5, \\
&\qquad\qquad y_1 + y_2 \le 4, \\
&\qquad\qquad x_1, x_2, y_1, y_2 \ge 0.
\end{aligned}$$

Using KKT conditions following problem is obtained:

$$\max_{x} \; x_1 + x_2 + 3y_1 - y_2$$

$$\begin{aligned}
s.t \quad & x_1 + x_2 + y_1 + y_2 + w_1 = 12, \\
& -x_1 + x_2 + w_2 = 2, \\
& 3x_1 - 4y_2 + w_3 = 5, \\
& y_1 + y_2 + w_4 = 4, \\
& 4x_1 + 8x_2 - 4y_1 + 2y_2 - u_1 - u_4 = -5 \\
& -4x_2 + 10y_2 - u_1 + 4u_3 - u_4 = -8 \\
& u_1 w_1 + u_2 w_2 + u_3 w_3 + u_4 w_4 = 0 \\
& v_1 y_1 + v_2 y_2 = 0 \\
& x_1, x_2, y_1, y_2, v_1, v_2, u_i, w_i \geq 0 \;, i = 1,2,3,4.
\end{aligned}$$

It is easy to show that by relaxing the u and v variables (by fixing them on zero or one) in the main problem, we can obtain upper bounds for the problem which might be not promising as expected. By enumeration of possible relaxation the best upper bound is shown in Table 1.

$$w_1 = w_2 = w_3 = w_4 = 0 \;, \; v_1 = v_2 = 0 \;\Rightarrow\; z^* = 23.$$

In genetic algorithm, the initial population is created according to the proposed rules in section 4. Also the best solution is produced by the following chromosome:

001100

According to this chromosome and above rules, we have the following results:

$$u_1 = u_2 = 0 \;, \; w_3 = w_4 = 0 \;, \; v_1 = v_2 = 0.$$

According to the Table 1, the best solution by the proposed algorithm equals to the optimal solution exactly. It can be seen that the proposed method is efficient and feasible from the results.

Table 1: comparison the best solutions - Example 1

Best solution by GA-LQBP/GA-LFBP	Optimal solution by references [16, 20]	Relaxation upper bound
$(x_1^*, x_2^*, y_1^*, y_2^*)$		

	z^*	$(x_1^*, x_2^*, y_1^*, y_2^*)$	z^*	$(x_1^*, x_2^*, y_1^*, y_2^*)$	z^*
(6.31, 1.68, 4, 0)	19.99	(6.312, 1.687, 4, 0)	20	(7.25, 2.23, 4.51, 0)	23.01

Example 2:

The following problem is linear fractional bi-level programming problem [13].

$$\begin{aligned}
&\max_x \quad \frac{5-2x-y}{2+x+y} \\
&\quad s.t \ \max_y \quad y \\
&\qquad s.t \quad -5x-3y \le -15, \\
&\qquad\qquad -x+4y \le 28, \\
&\qquad\qquad 2x+3y \le 32, \\
&\qquad\qquad 2x+2y \le 26, \\
&\qquad\qquad 2x-y \le 13, \\
&\qquad\qquad x-4y \le 3, \\
&\qquad\qquad x, y \ge 0.
\end{aligned}$$

Appling KKT conditions the above problem convert to this problem:

$$\begin{aligned}
&\max_x \quad \frac{5-2x-y}{2+x+y} \\
&s.t \quad -5x-3y+w_1 = -15, \\
&\qquad -x+4y+w_2 = 28, \\
&\qquad 2x+3y+w_3 = 32, \\
&\qquad 2x+2y+w_4 = 26, \\
&\qquad 2x-y+w_5 = 13, \\
&\qquad x-4y+w_6 = 3, \\
&\qquad -3u_1+4u_2+3u_3+2u_4-u_5-4u_6-v = -3, \\
&\qquad w_1u_1+w_2u_2+w_3u_3+w_4u_4+w_5u_5+w_6u_6 = 0, \\
&\qquad yv = 0, \\
&\qquad x, y, v, w_i, u_i \ge 0 \ , \ i = 1,\dots,6.
\end{aligned}$$

By enumeration of possible relaxation, the best upper bound is

$$w_1 = w_2 = w_3 = w_4 = 0, \ u_5 = u_6 = 0 \ , \ v = 0 \quad \Rightarrow \ z^* = 1.95$$

In genetic algorithm the initial population is created according to the proposed rules in section 4. Also the best solution is produced by the following chromosome:

0011100

Choosing $u_1 = u_2 = u_6 = 0$, $w_3 = w_4 = w_5 = 0$, $v = 0,$ by the proposed genetic algorithm, the optimal solution is obtained. The best solution is $(x^*, y^*) = (8.66, 4.33)$ and the upper level's objective function is 1.66 also the lower level's objective function is 8.66. The results are all close to the exact values in Ref [12, 21]. It is easy to see that the GA-LQBP/GA-LFBP algorithm is feasible according to the results. More computations are provided in Table 2. We use personal computer and MATLAB 7.1 for all our computations.

Table 2: comparison optimal solutions and elapsed time with deferent sizes of BLPP

Prob. No.	#Constraints	#Variables	Optimal Solution	Best Solution by BLGA		Gap by GA-LQBP	Best Solution by reference [17]		Gap by reference
1	4	2	-12	-12	0.15	0	-12.01	1.12	0.08 %
2	3	3	8.44	8.44	0.25	0	8.38	2.31	0.07 %
3	5	5	----	-31.3	1.35	----	----	----	----
4	6	10	----	2.5	3.5	----	----	----	----

7. Conclusion

We presented a genetic method for solving linear-quadratic bi-level programming and linear-fractional bi-level programming problems. Using the KKT conditions LQBP and LFBP were converted into single level problems. Then the problems were made simpler to linear programming by the chromosome according to the rule. Finally the algorithm finds the best solution better than references. In fact the GA-LQBP/GA-LFBP algorithm is a novel assay for solving two important forms of bi-level programming problem.

8 References

[1] J.F. Bard, Some properties of the bilevel linear programming, Journal of Optimization Theory and Applications, 68 371–378, 1991.

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

[3] Lv. Yibing, Hu. Tiesong, Wang. Guangmin , A penalty function method Based on Kuhn–Tucker condition for solving linear bilevel programming, Applied

Mathematics and Computation, 188 808–813, 2007.

[4] G. B. Allende, G. Still, Solving bi-level programs with the KKT-approach, Springer and Mathematical Programming Society 131:37 – 48, 2012.

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

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

[7] S. Pramanik, T.K. Ro, Fuzzy goal programming approach to multilevel programming Problems, European Journal of Operational Research 194 368–376, 2009.

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

[9] J. Nocedal, S.J. Wright, Numerical Optimization, Springer-Verlag, New York, 2005.

[10] A.AL KhayyalMinimizing a Quasi-concave Function Over a Convex Set: A Case Solvable by Lagrangian Duality, proceedings, I.E.E.E. International Conference on Systems, Man,andCybemeties, Tucson AZ 661-663, 1985.

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

[12] G. Wang, B. Jiang, K. Zhu, Global convergent algorithm for the bi-level linear fractional-linear programming based on modified convex simplex method, Journal of Systems Engineering and Electronics 239–243, 2010.

[13] W. T. Wend, U. P. Wen, A primal-dual interior point algorithm for solving bi-level Programming problems, Asia-Pacific J. of Operational Research, 17,2000.

[14] N. V. Thoai, Y. Yamamoto, A. Yoshise, Global optimization method for solving Mathematical programs with linear complementary constraints, Institute of Policy and Planning Sciences, University of Tsukuba, Japan 978, 2002.

[15] S.R. Hejazi, A. Memariani, G. Jahanshahloo, Linear bi-level programming Solution by genetic algorithm, Computers & Operations Research 29 1913–1925, 2002.

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

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

[18] B. Baran Pal, D .Chakraborti , P. Biswas, A Genetic Algorithm Approach to Fuzzy Quadratic Bi-level Programming, Second International Conference on Computing, Communication and Networking Technologies, 2010.

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

[20] H.X. Yin, R.S. Wang, Q. Wang, S.Y. Wang, Linear-quadratic bi-level programming, Systems Engineering- Theory & Practice (4) 1–8, 1994.

[21] H. I. Calvete, C. Cale, A penalty method for solving bi-level linear fractional/linear

programming problems. Asia-Pacific Journal of Operational Research 207–224, 2004.

[22] W.Zhongping, W.Guangmin, A Dual-Relax Penalty Function Approach for Solving Non- Linear Bi-Level Programming with Linear Lower Level Problem, Acta MathematicaScientia,31B(2):652–660, 2011.

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

[24] S. Dempe, A.B. Zemkoho, On the Karush–Kuhn–Tucker reformulation of the bi-level optimization problem, Nonlinear Analysis 75 1202–1218, 2012.

[25] Sh. Parisa, R. Tavakkoli, Solving a multi-objective multi-skilled manpower scheduling modelby a fuzzy goal programming approach, Applied Mathematical Modelling 37 5424–5443, 2013.

[26] S. Masatoshi, Takeshi.M, Stackelberg solutions for random fuzzy two-level linear Programming through possibility-based probability model, Expert Systems with Applications 39 10898–10903, 2012 .

[27] J. Yan, Xuyong.L, Chongchao.H, Xianing.W, Application of particle swarm Optimization based on CHKS smoothing function for solving nonlinear bi-level Programming problem, Applied Mathematics and Computation 219 4332–4339, 2013 .

[28] R.J. Kuo, Y.S. Han, A hybrid of genetic algorithm and particle swarm optimization For solving bi-level linear programming problem – A case study on supplychain model,
Applied Mathematical Modelling 35 3905–3917, 2011.

[29] G. Wanga, K.Zhua, An approximate programming method based on the simplex Method for bi-level programming problem, Computers and Mathematics with Applications
59 3355_3360, 2010.

[30] G. Zhang, J. Lu, Model, solution concept, and Kth-best algorithm for linear Tri-Level programming, Information Sciences 180 481–492, 2010.