AMO - Advanced Modeling and Optimization, Volume 15, Number 3, 2013

A Genetic Approach for Solving Bi-Level Programming Problems

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

[a]Ph.D. Candidate at Payamenur University of Tehran, Department of Mathematics, Tehran, Iran

[b]Professor of Industrial Engineering at University of Kurdistan, Department of Industrial Engineering, Sanandaj. Iran

Abstract

The bi-level programming problem (BLPP) however theoretically is interested by many researchers but in recent years it is known as 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 literature shows a few attempts for using meta-heuristic algorithms and in this paper 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 BLPP into single level problem. Thus by using the proposed GA, the single problems are solved. The proposed approach achieves ficient and feasible solutions and they are evaluated by comparing with references and test problems.

Keyword: The linear bi-level programming problem, Genetic algorithm, Karush-Kuhn–Tucker conditions.

1. Introduction

[*] **Corresponding author**

Email address: **eghbal_math@yahoo.com**

AMO - Advanced Modeling and Optimization. ISSN: 1841-4311

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 one is called follower. 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 solution [1, 2]. Nonetheless the BLPP is an applicable problem and practical tool to solve decision making problems. It is used in several areas such as economic, traffic, finance and so on.Therefore finding the optimal solution has a special importance to researchers.

Several algorithms have been presented for solving the BLPP [3, 4, 11, 12, 13]. These algorithms are divided into the following classes: Transformation methods [3, 4], Fuzzy methods [5, 6, 7, 8], Global techniques [9, 10, 11, 12], Primal–dual interior methods [13], Enumeration methods [14], Meta heuristic approaches [15, 16, 17, 18, 19].

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.We present a procedure based on genetic algorithm to solve these two problems. In the remaining of pages, in section 2, basic concepts of LQBP andLFBP are proposed. We provide the GA to solve LQBP and LFBP in section 3. Section 4 describes the steps of presented algorithm. Computational results are proposed in Section 5 and finally, the paper is finished in section 6.

2. The concepts and properties

We research 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]:

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

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), 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

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

The LFBP problem is formulated as follows [21]:

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

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

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

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

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

Based on the above assumptions the follower rational reaction set is

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

Where the inducible region is as follows

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

Finally the bi-level programming problem can be written as

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

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

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

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

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

3. Genetic algorithm to solve BLPP

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 [11, 15, 16] 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.

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:

$Algorithm\ 1: GA\ to\ solve\ BLPP$

$$
\begin{aligned}
&1: t = 0\\
&2: \text{initialize } P(t)\\
&3: \text{evaluate } P(t)\\
&4: \text{While not } \textit{terminate} \text{ do}\\
&5: \quad P'(t) = \text{recombine } P(t)\\
&6: \quad P''(t) = \text{mutate } P'(t)\\
&7: \quad \text{evaluate } P''(t)\\
&8: \quad P(t+1) = \text{select } (P''(t) \cup Q)\\
&9: \quad t = t+1\\
&10: \text{End of While}\\
&11: \text{End.}
\end{aligned}
\tag{11}
$$

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

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

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

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 theorem 2 was proposed by [12].

4. 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 S to the LQBP and LFBP respectively:

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

$$\begin{aligned} \max_y \quad & d^T y \\ s.t \quad & Ax + By \le r, \\ & x, y \ge 0. \end{aligned} \tag{15}$$

.Where r 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.

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. The best produced solution that has been recorded in the algorithm is reported as the best solution to BLPP by proposed GA algorithm.

Computational results

Two following examples are solved by use of the genetic algorithm proposed in this article to illustrate the feasibility and efficiency of the proposed algorithm. The first example is LQBP and the second example is LFBP.

Example 1:

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

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

Using KKT conditions following problem is obtained:

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

It is easy to show that by relaxing the u’s 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.

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

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:

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

Best solution by our method		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, 4.00, 0.00)	19.99	(6.31250, 1.68750, 4.0000, 0.0000)	20	(7.25, 2.23, 4.51, 0.00)	23.01

Table 1 comparison the best solutions - Example 1

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.

Example 2:

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

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

Appling KKT conditions the above problem convert to this problem:

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

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 $(x^*,y^*)=(8.66,4.33)$ is 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 proposed genetic algorithm is feasible according to the results.

6. 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 are converted into single level problems. Then the problems are made simpler to linear programming by the chromosome according to the rule.

Finally the algorithm presents the best solution better than references. In fact the proposed algorithm is a novel assay for solving two important forms of bi-level programming problem.

References

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

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

[3] Lv. Yibing, Hu.Tiesong, Wang.Guangmin,(2007)A penalty function method Based on Kuhn–Tucker conditionfor solving linear bilevel programming, Applied Mathematics and Computation 188 808–813.

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

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

[6]SSinha,(2003)Fuzzy programming approach to multi-levelprogramming problems, Fuzzy Sets And Systems 136 189–202.

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

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

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

[10] A.AL Khayyal,(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.

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

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

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

[14] N. V. Thoai,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.

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

[16]G. Z. Wang, 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.

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

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

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

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

[21] H. I. Calvete, C. Cale,(2004). A penalty method for solving bi-level linear fractional/linear programming problems. Asia-Pacific Journal of Operational Research207–224.