Journal of Scientific Research & Reports
6(7): 549-559, 2015; Article no.JSRR.2015.180
ISSN: 2320-0227

SCIENCEDOMAIN *international*
www.sciencedomain.org

Combining a Continuous Search Algorithm with a Discrete Search Algorithm for Solving Non-linear Bi-level Programming Problem

Eghbal Hosseini[1*] and Isa Nakhai Kamalabadi[2]

[1]Department of Mathematics, Payame Noor University of Tehran, Tehran, Iran.
[2]Department of Industry, University of Kurdistan, Sanandaj, Iran.

Authors' contributions

This work was carried out in collaboration between both authors. Author EH designed the study, wrote the protocol, and wrote the first draft of the manuscript. Author INK managed the literature searches, analyses of the study performed the spectroscopy analysis and author EH managed the experimental process and identified the species of plant. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JSRR/2015/15831
Editor(s):
(1) Ming-Jyh Chern, Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taiwan.
(2) Pak Kin Wong, Department of Electromechanical Engineering, University of Macau, Macao.
Reviewers:
(1) Anonymous, Brazil.
(2) Anonymous, China.
(3) Rajesh Chandrakant Sanghvi, Mathematics Department, G H Patel College of Engg. and Tech., V V Nagar, Gujarat Technological University, Gujarat, India.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=968&id=22&aid=8756

Original Research Article

Received 20th December 2014
Accepted 17th March 2015
Published 10th April 2015

ABSTRACT

The multi-level programming problems, have received much interest from researchers because of their application in several areas such as economic, traffic, finance, management, transportation and so on. Among these, the bi-level programming problem (BLPP) is an appropriate tool to model these real problems. It has been proven that the general BLPP is an NP-hard problem, so it is a practical and complicated problem therefore solving this problem would be significant. However the literature shows several algorithms to solve different forms of the bi-level programming problems (BLPP), but there is no any hybrid approach of combining of two meta-heuristic algorithms. In this paper, the authors combine particle swarm optimization (PSO), which is a continuous approach, with a proposed modified genetic algorithm (MGA), which is a discrete algorithm, using a heuristic function and constructing an effective hybrid approaches (PSOMGA). Using the Karush-Kuhn-

**Corresponding author: Email: eghbal_math@yahoo.com;*

Tucker conditions the BLPP is converted to a non-smooth single level problem, and then it is smoothed by a new heuristic method for using PSOMGA. The smoothed problem is solved using PSOMGA which is a fast approximate method for solving the non-linear BLPP. The presented approach achieves an efficient and feasible solution in an appropriate time, as justified by comparison with test problems.

Keywords: Particle swarm optimization; genetic algorithm; non-linear bi-level programming problem; karush-kuhn-tucker conditions.

NOMENCLATURE

$F_1(x,y,z)$	*Objective function of the first level in the TLPP*
$F_2(x,y,z)$	*Objective function of the second level in the TLPP*
$F_3(x,y,z)$	*Objective function of the third level in the TLPP*
$g(x,y,z)$	*Constraints in the TLPP*
w	*Slack variable*
v	*Slack variable*
$F(x,y)$	*Objective function of the first level in the BLPP*
$f(x,y)$	*Objective function of the first level in the BLPP*
$g(x,y)$	*Constraints in the BLPP*
S	*A nonempty convex set*
L	*Lagrange function*
α	*Lagrange Coefficient*
β	*Lagrange Coefficient*
μ	*Lagrange Coefficient*
P	*Initial population*
P'	*Crossover population*
P''	*Mutation population*
Q	*Set of chromosomes in the current generation*
(x^*,y^*,z^*)	*Optimal solution for the TLPP*
(x^*,y^*)	*Optimal solution for the BLPP*

1. INTRODUCTION

It has been proven that the bi-level programming problem (BLPP) is an NP-Hard problem [1,2]. Several algorithms have been proposed to solve BLPP [3,4,5,6,7,8,9,10,11,12]. These algorithms are divided into the following classes: global techniques [13,14,15,16], enumeration methods [17], transformation methods [18,19,20,21], meta heuristic approaches [22,23,24,25,26,27,28], fuzzy methods [29,30,31], primal-dual interior methods [5]. In the following, these techniques are shortly introduced.

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. A summary of important properties for convex problem are as follows, which $F: S \to R^n$ and S is a nonempty convex set inR^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 Fover$S$.

The BLPP is used frequently by problems with decentralized planning structure. It is defined as [32]:

$$\begin{aligned} &\min_x F(x,y) \\ &s.t \min_y f(x,y) \\ &s.t g(x,y) \le 0, \\ &x,y \ge 0. \end{aligned} \tag{1}$$

Where

$$F: R^{n\times m} \to R^1, f: R^{n\times m} \to R^1,$$
$$g: R^{n\times m} \to R^q, x \in R^n, y \in R^m.$$

Also F and f are objective functions of the leader and follower respectively.

However, there are meta– heuristic approaches and their combinations to solve optimization problems [25], but there is no any approach which combines a continuous algorithm and a discrete one in these approaches. In this paper, the authors have tried to combine particle swarm optimization, a continuous approach, and proposed modified genetic algorithm, a discrete algorithm, to solve non-linear BLPP.

The remainder of the paper is structured as follows: problem formulation and smoothing method to the BLPP are introduced in Section 2. The algorithm based on combining modified genetic algorithm and particle swarm optimization is proposed in Section 3. Computational results are presented for our approaches in Section 4. As result, the paper is finished in Section 5 by presenting the concluding remarks.

2. PROBLEM FORMULATION AND SMOOTHING METHOD

The feasible region of the non-linear BLPP is

$$S = \{(x,y) | g(x,y) \leq 0, x, y \geq 0\} \quad (2)$$

Using KKT conditions Equation (1) can be converted into the following problem:

$$\begin{aligned} &\min_{x,y\mu} F(x,y,\mu) \\ &s.t \nabla_y L(x,y,\mu) = 0, \\ &\mu g(x,y) = 0, \\ &g(x,y) \leq 0, \\ &\mu, x, y \geq 0. \end{aligned} \quad (3)$$

Where L is the Lagrange function and $L(x,y,\mu) = f(x,y) + \mu g(x,y)$.

To convert the inequality constraint to an equality constraint, the positive slack variable v is added:

$$\begin{aligned} &\min_{x,y\mu} f(x,y,\mu) \\ &s.t \nabla_y L(x,y,\mu) = 0, \\ &\mu g(x,y) = 0, \\ &g(x,y) + v = 0, \\ &x, y, v, \mu \geq 0. \end{aligned} \quad (4)$$

Let $\alpha = g(x,y)$ then the problem can be written as follows:

$$\begin{aligned} &\min_{x,y\mu} f(x,y,\mu) \\ &s.t \nabla_y L(x,y,\mu) = 0, \\ &\mu\alpha = 0, \\ &\alpha + v = 0, \\ &x, y, \mu, v \geq 0. \end{aligned} \quad (5)$$

2.1 Modified Genetic Algorithm (MGA)

In this section, a modified genetic algorithm is proposed then basic and general concepts related to particle swarm optimization algorithms are discussed. Finally, the hybrid method of both algorithms is proposed.

Genetic algorithms are global methods which 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.
6. These algorithms are inherently discrete.

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.

In the suggested method, every chromosome is demonstrated by a string. This string consists of$\mathrm{k} + \mathrm{l} + 2\mathrm{p}$corresponding variables $x, y, \mu, v,$ also these chromosomes are applied in Equation (5) 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, Equation (5) is prepared for using genetic algorithm:

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

If the *i-th* component of the chromosome is equal to zero, then $\alpha_i = 0, v_i \geq 0$ Else $\alpha_i \geq 0, v_i = 0$.
If the *j-th* component of the chromosome is equal to zero, then $\alpha_j = 0, \mu_j \geq 0$ Else $\alpha_j \geq 0, \mu_j = 0$.
Theorem 3.1:

(x^*, y^*) is the optimal solution to the Equation (1) if and only if there exists such that $(x^*, y^*, \alpha^*, \mu^*, v^*)$ is the solution of the Equation (5).

Proof :

The proof of this theorem was given by [17].

The MGA steps are proposed as follows:

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:

$$\begin{aligned} &\min_y f(x,y) \\ &s.tg(x,y) \leq 0, \\ &x,y \geq 0. \end{aligned} \quad (6)$$

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,…,k+l). The operation sum is defined as follows:

$$X_i + Y_i = \left[\frac{x_i + y_i}{2}\right]$$

That $X_i, Y_i$is i-th component of chromosomesX, Y. The other components are remained the same as the first parent.

2. The (k+l+i)-th component of the second child is replaced by the sum of the (k+l+i)-th components of parents i=1,2,…,2p. 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=3we generate the following children:

Parents	Children
92745 1036 786 123	54565 0057 786 123
27386 0178 193 321	27386 0178 484 222

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 according following function:

$$M(i) = \begin{cases} i+1 & if\, i \neq 9 \\ 0 & if\, i = 9 \end{cases} \quad (7)$$

In fact if the value of the chosen gene be i, it will be changed to i+1 which $i \neq 9$ and if the value of the chosen gene be 9, it will be changed to 0.

For example, by applying the present mutation operation to the following chromosome, and k= 5, l=4, p=3, we have:

Before mutation	After mutation
17834 9267 193 052	28945 0378 204 163

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 TLPP by proposed GA algorithm.

2.2 Particle Swarm Optimization (PSO)

Swarm intelligence system is an artificial intelligence technique which is usually made up of a population of simple particles cooperation locally with one another and with their environment.

Because PSO simply solves discontinuous and non-convex problems, therefore it is suitable tool for solving BLPP. It is a method based on population search. In each iteration, PSO moves from a set of particle positions to the better one set with improvement optimal solution. It is inherently continuous. PSO has the following steps:

1. The initial population of particles and their velocity are produced randomly in the following feasible region.
2. The best objective function for each particle in each iteration is kept. Also the global best objective function is defined.
3. The global best and the best objective function for each particle are updated as follows:

$$v^i[t+1] = wv^i[t] + c_1r_1(x^{ibest}[t] - x^i[t]) + c_2r_2(x^{gbest}[t] - x^i[t]) \quad (8)$$
$$x^i[t+1] = x^i[t] + v^i[t+1]$$

4. If termination conditions are not satisfied, the above steps will be continued from step two. Otherwise the algorithm will be finished.

2.3 Hybrid Algorithm by Combining PSO and MGA (PSOMGA)

As mention previously, the genetic algorithm searches in the discrete space but the particle swarm optimization searches in the continuous one. Therefore to hybrid these two approaches it is necessary that we round the positions of particles, which are continuous, to use genetic algorithm.

In this method initial population is produced using PSO algorithm. Then position and their velocity are updated by the step 3 in PSO and the new population with better objective functions will be made. After that, the position of particles will be rounded then MGA is applied to the new population by these steps: crossover operation, mutation and selection. In the next iteration PSO algorithm is applied to the obtained population by MGA. The algorithm is continued while termination condition is satisfied. In fact in this proposed hybrid method by combining PSO and MGA the genetic algorithm is used after the particle swarm optimization algorithm. Therefore the convertor function should be defined to convert particles to the acceptable points for MGA (chromosome).

2.3.1 Definition 3.1

We define the convertor function to round an arbitrary number x as follows:

$$INT^*(x) = \begin{cases} n & if x < n + \frac{1}{2} \\ n+1 & if x \geq n + \frac{1}{2} \end{cases}, \quad (9)$$

For $x \in R$. Also we define for $x \in R^K, INT^*(x) = (INT^*(x_1), INT^*(x_2), \dots, INT^*(x_k))$.

Example 1:

To more illustrate the proposed function that converts the particles to the point using GA, consider $X_1 = (0.4,0.44), X_2 = (0.7,0.5), X_3 = (0.3,0.6), X_4 = (0.9,0.2), X_5 = (1.6,1.3)$.

$$INT^*(X_1) = (0,0)$$
$$INT^*(X_2) = (1,1)$$
$$INT^*(X_3) = (0,1)$$
$$INT^*(X_4) = (1,0)$$
$$INT^*(X_5) = (2,1)$$

These points have been shown in Fig. 1. The green points are the produced particles by PSO and the red points are the acceptable point for GA (chromosome). It is easy to see that, each green point is attracted to the nearest red point by the proposed convertor function.

Now everything is prepared to propose the new hybrid approach. The algorithm steps as follows:

Step 1: initialization

The initial population of particles and their velocity are produced randomly in the Equation (8).

Step 2: Keeping the present best particles in an array

The best objective function for each particle is

kept in the array at the each iteration. Also the global best objective function is saved. This process continues till the algorithm is finished.

Step 3: Updating

The global best and the best position for each particle are updated according to (8).

Step 4: Checking the optimal solution

If $\mathrm{d}(\mathrm{F}(x^{gbest}[t+1]), \mathrm{F}(x^{gbest}[t])) < \varepsilon_1$ then go to step 8. Otherwise go to the next step.

Step 5: Converting particles to chromosome

In this step all particles convert to chromosome using proposed convertor function.

Step 6: Crossover operation

In this step each particle is correspond with a chromosome. The chromosomes of the current population are arranged in descending order of objective function values. Then numbers of the best chromosomes are selected to use crossover operation for positions by the proposed rules in step 3 of section 3.1. The velocities are changed too as follows:

$$Minus = c_1 r_1 v_1^i[t+1] - c_2 r_2 v_2^i[t+1]$$
$$Sum = c_1 r_1 v_1^i[t+1] + c_2 r_2 v_2^i[t+1]$$
$$v_1^i[t+1] = Minus$$
$$v_2^i[t+1] = Sum$$

Which v_1^i, v_2^i are respectively the velocities of first and second particles in the *i-th*iteration.

Step 7: Mutation

In this step each chosen gene of every chromosome, mutates like step 4 of section 3.1 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.

Also the velocities are changed according to the following rule:

$$v^i[t+1] = v^i[t+1] + \frac{\pi}{2}$$

Step 8: Selection

We select a new population similar to the size of the first generation by combing the obtained particles in two above steps and number of the particles in current population before applying MGA.

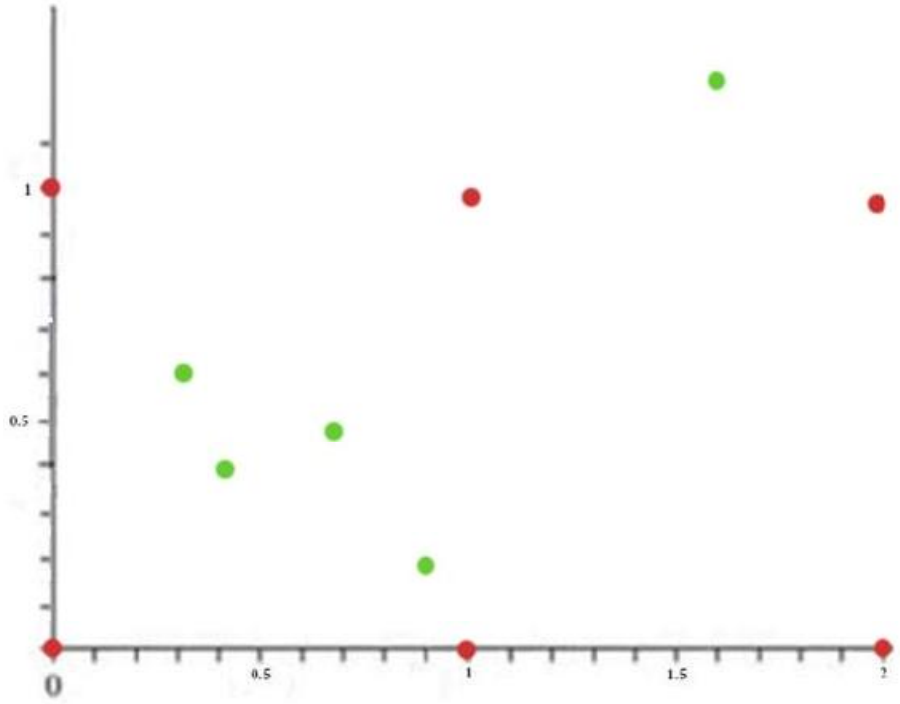


Fig. 1. Converting particles to chromosomes for example1

Step 9: Termination

If $\mathrm{d}(\mathrm{F}(x^{gbest}[t+1]), \mathrm{F}(x^{gbest}[t])) < \mathcal{E}_1$ then the algorithm is finished and $x^{gbest}[t+1]$ is the best solution by the proposed algorithm. Otherwise, let k=k+1 and go to the step 2. That d is the following metric:

$$\mathrm{d}\big(\mathrm{F}(x^{gbest}[t+1]), \mathrm{F}(x^{gbest}[t])\big) = (\sum_{i=1}^{k+l+2p} (\mathrm{F}({x_i}^{gbest}[t+1]), \mathrm{F}({x_i}^{gbest}[t]))^2)^{\frac{1}{2}}.$$

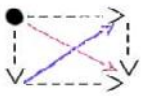
Because the authors going to gain the best optimal solution, the Termination condition, $\mathrm{d}(\mathrm{F}(x^{gbest}[t+1]), \mathrm{F}(x^{gbest}[t])) < \mathcal{E}_1$, will be checked in bothsteps 4 and 9. In fact this condition will be checked before and after MGA which rounds the solutions. Also both of steps crossover and mutation in the MGA are applied to the velocities of particles. To illustrate these two operations following example is proposed.

Example 2:

Consider following particles and their velocities that are the parents in MGA:

Using step 6 in the proposed hybrid algorithm after mutates the velocities are changed as follows:

Also after using step 5 velocities of children are changed that the red direction is resultant of two directions of parents and the blue one is difference of them.



Theorem 3.1 Sequence$\{F_k\}$ which was proposed in above algorithm is convergent to the optimal solution, so that the algorithm is convergent.

Proof:

Let $(F_l) = (F(t^l)) = (F(t_1^l), F(t_2^l), \dots, F(t_{n+2m}^l))$= $(F_1^{(l)}, F_2^{(l)}, \dots, F_{n+2m}^{(l)})$.

According to step 4

$$d(F_{k+1}, F_k) = \mathrm{d}\left(\mathrm{F}(\mathrm{t}^{\mathrm{k+1}}), \mathrm{F}(\mathrm{t}^{\mathrm{k}})\right) = (\sum_{i=1}^{n+2m} (F(t_i^{k+1}) - F(t_i^k))^2)^{\frac{1}{2}} < \varepsilon_1$$

Therefore $(\sum_{i=1}^{n+2m} \left(F(t_i^{k+1}) - F(t_i^k)\right)^2) < {\varepsilon_1}^2$

There is large number such as N which k+1>k>N and j=1,2,…,2m+n we have:

$(F_j^{(k+1)} - F_j^{(k)})^2 < {\varepsilon_1}^2$, therefore $\left|F_j^{(k+1)} - F_j^{(k)}\right| < \varepsilon_1$ (10)

$Nowletm = k+1, r = kthenwehave$

$\forall_{m>r>N} |F_j^{(m)} - F_j^{(r)}| < \varepsilon_1.$

This shows that for each fixed j, $(1 \le \mathrm{j} \le 2\mathrm{m}+\mathrm{n})$, the sequence $(\mathrm{F}_\mathrm{j}^{(1)}, \mathrm{F}_\mathrm{j}^{(2)}, \dots)$ is Cauchy of real numbers, then it converges by theorem 3.2.

Say, $F_j^{(m)} \to F_j$ as $m \to \infty$. Using these 2m+n limits, we define $F = (F_1, F_2, \dots, F_{2m+n})$. From (10) and m=k+1, r=k,

$$d(F_m, F_r) < \varepsilon_1$$

Now if $\mathrm{r} \to \infty$, by$F_r \to \mathrm{F}$ we have $d(F_m, F) \le \varepsilon_1$.

This shows that F is the limit of (F_m) and the sequence is convergent therefore proof of theorem is finished.

3. COMPUTATIONAL RESULTS

To illustrate the algorithm, we first propose the practical following examples and then model them. Finally the proposed examples will be solved using our algorithm.

Example 3 [11] Consider the following linear BLPP where $x \in R^1, y \in R^1$.

Consider the following linear bi-level programming problem:

$$\begin{aligned}
&\min_{\mathrm{x}} x^2 + (\mathrm{y}-10)^2 \\
&\text{s.t} \\
&\min_{\mathrm{y}} (x + 2y - 30)^2 \\
&\quad \text{s.t} \\
&\mathrm{x} - y^2 \ge 0, \\
&\quad 20 - \mathrm{x} - y^2 \ge 0, \\
&0 \le \mathrm{x} \le 15, \\
&y \ge 0.
\end{aligned}$$

Using KKT conditions the following problem is obtained:

$$\begin{aligned}
&\min_{x} x^2 + (y-10)^2 \\
&\text{s.t} \\
&4(x+2y-30)=0, \\
&2y(\mu_1+\mu_2)=0, \\
&\mu_1(y^2-x)=0, \\
&\mu_2(x+y^2-20)=0, \\
&\mu_3(x-15)=0, \\
&y^2-x\le 0, \\
&\quad x+y^2-20\le 0, \\
&y,\mu_1,\mu_2,\mu_3\ge 0.
\end{aligned}$$

We solve this problem using the PSOMGA algorithm and we present the optimal solution in the Table 1. Behavior of variables is shown in Fig. 2.We have present behavior of the x and optimal solution (OS) with different value of α,μ,v in Fig. 3.

Example 4 [11] Consider the following linear BLPP where $x \in R^1, y \in R^2$.

Consider the following linear bi-level programming problem.

$$\begin{aligned}
&\min_{x} x_1^2 - 2x_1 + x_2^2 - 2x_2 + y_1^2 + y_2^2 \\
&\text{s.t} \\
&\min_{y} y_1^2 - 2x_1y_1 + y_2^2 - 2x_2y_2 \\
&\quad \text{s.t} \\
&0.25-(y_1-1)^2\ge 0, \\
&0.25-(y_2-1)^2\ge 0, \\
&x_1,x_2,y_1,y_2\ge 0.
\end{aligned}$$

After applying KKT conditions and smoothing method, and then proposed PSOMGA algorithm above problem will be solved. The optimal solution is obtained using our method according to the Table 1.Behavior of variables is shown in Fig. 4. The behavior of the x and optimal solution (OS) with different value of α,μ,v in Fig. 5 have been shown.

More problems with different sizes have been solved by our approach and computation results have been proposed in Table 2. References of the examples in Table 1 are as follows:

Example 5 [3], Example 6 [7], Example 7 [26], Example 8 [27] which both of them are problems.

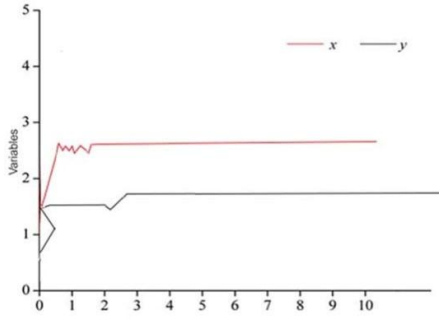


Fig. 2. Behavior of the variables in example 3

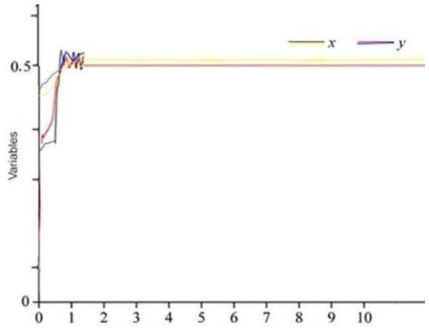


Fig. 3. Behavior of the variables in example 4

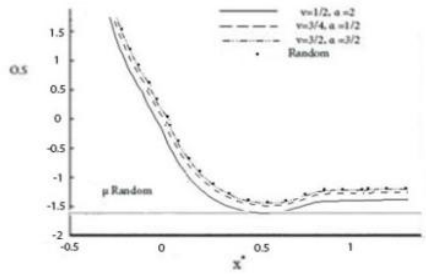


Fig. 4. Behavior of the x and optimal solution (OS) with different value of α, μ, vin example 3

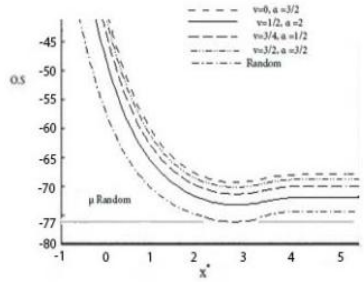


Fig. 5. Behavior of the x and optimal solution (OS) with different value of α, μ, v in example 4

Table 1. Comparison optimal solutions with deferent examples 5-8by PSOMGA

Example	Best solution by PSOMGA	Best solution reported in references	Optimal solution
3	(2.600,1.612)	(2.600,1.613) [30]	(2.600,1.612)
4	(0.51,0.51,0.51,0.51)	(0.5,0.5,0.5,0.5) [4]	(0.51,0.51,0.51,0.51)
5	(1.888,0.888,0.000)	(1.883,0.891,0.003) [3,7,26,27]	$(17/9, 8/9, 0)$
6	(0,0)	(0,0) [3,7,26,27]	(0,0)
7	(1,0)	(1,0) [3,7,26,27]	(1,0)
8	(0,0.75,0,0.5,0)	(0,0.75,0,0.5,0) [3,7,26,27]	(0,0.75,0,0.5,0)

Table 2. Comparison optimal solutions improvement by PSOMGA

	PSOMGA algorithm				
	Gap of optimal solution	Improvement rathar than [22,23,8,9,30]	Improvement rathar than [24]	Iterations	Time
Example 3	0	0.0001%	0.0003%	2000	0.47 s
Example 4	0	0.06%	0.005%	1500	0.56 s
Example 5	0	0.007%	0.001%	5300	2.42 s
Example 6	0	0 %	0%	2200	1.25 s
Example 7	0	0%	0%	3800	2.13 s
Example 8	0	0%	0.004%	5000	3.07 s

4. CONCLUSION AND FUTURE WORK

The main difficulty of the multi-level programming problem is that after using the KKT conditions the non-linear constraints are appeared. In this paper was attempted to remove these constraints by the proposed theorem, slack variables and proposed PSOMGA algorithm. As mentioned previously the authors have been combined two continuous and discrete effective approaches to the non-linear BLPP which this form of combining has not been studied by any researchers. According to the Tables the proposed method presents optimal solution in appropriate time and iterations. In the future works, the following should be researched:

(1) Examples in the larger sizes can be supplied to illustrate the efficiency of the proposed algorithm.
(2) Showing the efficiency of the proposed algorithm for solving other kinds of BLPP such as quadratic.
(3) Solving other kinds of multi-level programming problem such as tri-level programming problem.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Bard JF. Some properties of the bi-level linear programming. Journal of Optimization Theory and Applications. 1991;68 :371–378.
2. Vicente L, Savard G, Judice J. Descent approaches for quadratic bi-level programming. Journal of Optimization Theory and Applications. 1994;81:379–399.
3. Mathieu R, Pittard L, Anandalingam G. Genetic algorithm based approach to bi-level Linear Programming. Operations Research. 1994;28:1–21.
4. Wang G, Jiang B, Zhu K. Global convergent algorithm for the bi-level linear fractional-linear programming based on modified convex simplex method. Journal of Systems Engineering and Electronics. 2010;239–243.
5. Wend WT, Wen UP. A primal-dual interior point algorithm for solving bi-level programming problems, Asia-Pacific J. of Operational Research. 2000;17.
6. Yan J, XuyongL, Chongchao H, Xianing W, Application of particle swarm optimization based onCHKS smoothing function for solving nonlinear bi-level programming problem. Applied Mathematics and Computation. 2013;219:4332–4339.

7. Xu P, Wang L. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. Computers & Operations Research. 2014;41:309-318.
8. Wan Z, Mao L, Wang G. Estimation of distribution algorithm for a class of nonlinear bilevel programming problems. Information Sciences. 2014;256:184-196.
9. Zheng Y, Liu J, Wan Z. Interactive fuzzy decision making method for solving bi-level programming problem. Applied Mathematical Modelling. 2014;38(13): 3136-3141.
10. Zhang, G, Lu J, Montero J, Zeng Y, Model. solution concept, and Kth-best algorithm for linear tri-level. Programming Information Sciences. 2010;180:481–492.
11. Jiang Y, Li X, Huang C, Wu X. An augmented Lagrangian multiplier method based on a CHKS smoothing function for solving nonlinear bi-level programming problems. Knowledge-Based Systems. 2014;55:9-14.
12. He X, Li C, Huang T, Li C. Neural network for solving convex quadratic bilevel programming problems, Neural Networks. 2014;51:17-25.
13. Nocedal J, Wright SJ. Numerical optimization, Springer-Verlag, New York; 2005.
14. AL Khayyal A. 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. 1985;661-663.
15. Thoai NV, Yamamoto Y, Yoshise A. Global optimization method for solving mathematical programs with linear complementary constraints, Institute of Policy and Planning Sciences, University of Tsukuba, Japan. 2002;978.
16. Hejazi SR, Memariani A, Jahanshahloo G. Linear bi-level programming solution by genetic algorithm, Computers & Operations Research. 2002;29:1913–1925.
17. Lv. Yibing Hu. Tiesong, Wang. Guangmin, A penalty function method Based on Kuhn–Tuckercondition for solving linear bilevel programming. Applied Mathematics and Computation. 2007;188:808–813.
18. Allende GB, Still G. Solving bi-level programs with the KKT-approach. Springer and Mathematical Programming Society. 2012;1 31:37– 48.
19. Hosseini EE Nakhai I Kamalabadi. Line search and genetic approaches for solving linear tri-level programming problem. International Journal of Management, Accounting and Economics. 2014;1:4.
20. Hosseini E, Nakhai Kamalabadi I. Taylor approach for solving Non-linear Bi-level programming problem ACSIJ Advances in Computer Science: An International. 2014;3(5):11.
21. Arora SR, Gupta R, Interactive fuzzy goal programming approach for bi-level programming problem. European Journal of Operational Research. 2007;176:1151–1166.
22. Wang GZ, Wan X, Lv. Wang Y Genetic algorithm based on simplex method for solving Linear-quadratic bi-level programming problem. Computers and Mathematics with Applications. 2008;56: 2550–2555.
23. Hu TX, Guo X. Fu Y Lv. A neural network approach for solving linear bi-level programming problem. Knowledge-Based Systems. 2010;23:239–242.
24. Baran Pal B, Chakraborti D, Biswas P. A Genetic Algorithm Approach to Fuzzy Quadratic Bi-level Programming. Second International Conference on Computing, Communication and Networking Technologies; 2010.
25. Wan ZG, 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.
26. Hosseini E, I.Nakhai Kamalabadi, A Genetic Approach for Solving Bi-Level Programming Problems, Advanced Modeling and Optimization. 2013;15.
27. Hosseini E, Nakhai Kamalabadi I. Solving linear bi-level programming problem using two new approaches based on line search. International Journal of Management sciences and Education. 2014;2(6):243-252.
28. Sakava M, Nishizaki I, Uemura Y. Interactive fuzzy programming for multilevel linear programming problem.

Computers & Mathematics with Applications. 1997;36 71-86.

29. Sinha S. Fuzzy programming approach to multi-level programming problems. Fuzzy Sets and Systems. 2003;136:189-202.
30. Pramanik S, Ro TK, Fuzzy goal programming approach to multilevel programming problems. European Journal of Operational Research. 2009;194:368–376.
31. Hosseini EE, Nakhai Kamalabadi I. Two approaches for solving Non-linear Bi-level programming problem. Advances in Research. 2015;4:3. ISSN: 2348-0394.
32. Bard JF. Practical bi-level optimization: Algorithms and applications, Kluwer Academic Publishers, Dordrecht; 1998.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=968&id=22&aid=8756