

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com



Solving bi-level programming with multiple linear objectives at lower level using particle swarm optimization

Fatehem Matroud

*Islamic Azad University, Abadan branch, Department of mathematic, Abadan, Iran
fati.matroud@gmail.com*

Habibeh Sadeghi

*Department of Mathematics, Shahid Chamran University of Ahwaz, Ahwaz, Iran
habibe.sadeghi@mail.scu.ac.ir*

Article history:

Received April 2013

Accepted May 2013

Available online May 2013

Abstract

Bi-level programming is a tool for modeling decentralized decisions that consists of the objective of the leader at its first level and that of the follower at the second level. This paper deals with general bi-level optimization problems with multiple objectives at the lower level of decision making. We present Particle swarm optimization (PSO) algorithm for solving this problem. Also, two numerical examples are given to illustrate efficiency of the proposed algorithm.

Keywords: Bi-level optimization, Multi objective optimization, Particle Swarm Optimization.

1. Introduction

The bi-level programming problem is an optimization problem in which the constraints are implicitly determined by another optimization problem. In other words, it is a hierarchical optimization problem consisting of two levels. At the upper level, the decision maker (leader) has to choose first a strategy x_1 , and then lower level decision maker (follower) has to select a strategy x_2 that minimizes its own objective function parameterized by x_1 . Anticipating the reaction of the follower, the leader intends to find such values for its variables which together with the follower's reaction minimize its objective function.

General bi-level problems can be formulated as

$$\min_{(x_1, x_2)} \in S \quad f_1(x_1, x_2),$$

$$\text{where } x_2 \in \operatorname{argmin}_{v \in S(x_1)} f_2(x_1, v)$$

$x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ Are the variables controlled by the first level and second level decision makers, respectively. $S \subset \mathbb{R}^n$ define the common constraint region and $S(x_1) = \{x_2 \in \mathbb{R}^{n_2} : (x_1, x_2) \in S\}$. Due to their structure, bi-level programming problems are no convex and quite difficult to deal with. For an extensive bibliography the reader can refer to Demp (2003), Vicene and Calamai(1994).

In this study, we are concerned with bilevel programming problems where the lower level is a multiobjective optimization problem (BPMLO). Bi-level problems with multiple objectives at the lower level have been considered in the literature by Bonnel [3] that provides first order necessary conditions for the solution of the bilevel problem when dealing with weakly efficient and properly efficient solution of the lower level. Calvete and Gale [6] proved that the feasible region consists of faces of the polyhedron defined by the constraints. Then, by assuming that the upper level objective function is quasi concave, concluded that feasible region of this problem consists of faces of the polyhedron define by the constraints and there is an extreme point of this polyhedron which solves the problem. Finally, based on this property, two algorithm have presented for solving the problem. Particle swarm optimization (PSO) is an optimization algorithm proposed by Kennedy and Eberhart in 1995[12]. It is a relatively novel heuristic algorithm inspired by the choreography of a bird flock, which has been found to be quite successful in a wide variety of optimization tasks [11]. Due to its high speed of convergence and relative simplicity, the PSO algorithm has been employed by many researchers for solving bi-level linear programming problems. In this paper, a PSO algorithm is presented for solving BPMLO by focused on decreasing computational time.

The rest of the paper is organized as follows: In section 2, the problem formulation is provided. The proposed algorithm for solving bilevel programming with multiple linear objective functions at lower level is presented in section 3. In section 4, some numerical examples are given to demonstrate the proposed algorithm, while the conclusion is reached in section 5.

2 Bi-level programming with multiple linear objective functions at lower level

The bi-level programming with multiple linear objective functions at lower level problem can be formulated as:

$$\begin{aligned} \min_{x_1} \quad & f(x_1, x_2) \\ \text{s.t.} \quad & A_1^1 x_1 \leq b^1 \\ & x_1 \geq 0 \end{aligned} \tag{2.1}$$

where, x_2 solves

$$\begin{aligned} \min_{x_2} \quad & f(d_1 x_2, \dots, d_k x_2) \\ \text{s.t.} \quad & A_1^2 x_1 + A_2^2 x_2 \leq b^2 \\ & x_2 \geq 0 \end{aligned} \tag{2.2}$$

$x_1 \in R^{n_1}$ and $x_2 \in R^{n_2}$ are the vector of variables which controlled by the leader and follower respectively. $f: R^{n_1} \times R^{n_2} \rightarrow R$, $b^1 \in R^{m_1}$, $b^2 \in R^{m_2}$ and A_1^1, A_1^2, A_2^2 are matrices of suitable dimensions. Also, we introduce the following sets:

$$T = \{(x_1, x_2) \in R^{n_1} \times R^{n_2} : A_1^1 x_1 \leq b^1, A_1^2 x_1 + A_2^2 x_2 \leq b^2, x_1 \geq 0, x_2 \geq 0\}$$

$$T_1 = \{x_1 \in R^{n_1} : \exists x_2 \in R^{n_2} \text{ such that } (x_1, x_2) \in T\}$$

$$V = \{(x_1, x_2) \in R^{n_1} \times R^{n_2} : A_1^1 x_1 \leq b^1, x_1 \geq 0\}$$

$$S = \{(x_1, x_2) \in R^{n_1} \times R^{n_2} : A_1^2 x_1 + A_2^2 x_2 \leq b^2, x_2 \geq 0\}$$

$$S(x_1) = \{x_2 \in R^{n_2} : A_2^2 x_2 \leq b^2 - A_1^2 x_1, x_2 \geq 0\}$$

We assume T is a nonempty polyhedral. We also assume that the upper level objective function is continuous. In a multiobjective programming problem, several objective function have to minimized simultaneously. Usually, there is no single point will minimized all objective functions given at once. Therefore, we use the concept of efficiently or Pareto optimality. We recall that a lower level feasible solution $x_2 \in S(x_1)$ is an efficient solution of the lower level, if there is no $\tilde{x}_2 \in S(x_1)$ such that $d_i \tilde{x}_2 < d_i x_2$ for $i = 1, \dots, k$, and $d_j \tilde{x}_2 \leq d_j x_2$ for some $j = 1, \dots, k$. For linear multiobjective problem at the lower level, we show by $E(x_1)$ is the set of all pareto optimal points. Assuming that the set $S(x_1)$ is nonempty and compact, $E(x_1)$ is connected and consists of union of maximally efficient faces of $S(x_1)$ [6]. A face of $S(x_1)$ is called an efficient face if and only if it contains efficient solution in its relative interior. The feasible region of the BPMLO, called the inducible region (IR), is defined by $IR = \{(x_1, x_2) \in R^{n_1} \times R^{n_2} : (x_1, x_2) \in T, x_2 \in E(x_1)\}$. A point of the region IR is called a bilevel feasible solution. A point (x_1^*, x_2^*) is an optimal solution of the BPMLO, if (x_1^*, x_2^*) be the optimal solution of the following problem:

$$\begin{aligned} \min \quad & f(x_1, x_2) \\ \text{s.t.} \quad & (x_1, x_2) \in IR \end{aligned} \tag{2.3}$$

We consider the following definition and theorems [6].

Definition 2.1. Consider a face F_T of the polyhedron T . F_T is called a bilevel face if $F_T \subset IR$, that is to say, all its points be bilevel feasible solutions.

Theorem 2.1 If the region of IR is nonempty, the BPMLO has an optimal solution, the proofs can be found in [6].

Theorem 2.2 A face F_T of the polyhedron T is a bilevel face if and only if it has a bilevel feasible solution (\bar{x}_1, \bar{x}_2) in its relative interior.

Theorem 2.3 The region of IR consists of the union of faces of polyhedron T .

Theorem 2.4 Assuming that $f(x_1, x_2)$ is a quasiconcave function, there exists an extreme point of the polyhedron T which is an optimal solution of the BPMLO.

3. A Particle Swarm optimization for solving BPMLO

The bilevel multiobjective programming (BLMOP) has seldom been studied. A number of interactive algorithm for BLMOP are presented [1,13,14,16]. Calvete and Cale in [6] proposed the k-best and genetic algorithm for BPMLO, they solved test problems with genetic algorithm by focus on four factors: the probabilities of crossover and mutation, the population size and the number of iteration. By the review the results, we conclude that, the computation time increases rapidly when population size and number of iterations increase. In most cases PSO is more effective than traditional algorithms [13]. There is a great distinction between PSO and genetic algorithms. In genetic algorithms, chromosomes share the information, which causes the whole community moves gradually into a better area, while in PSO the information is sent out by the best particle which is followed by other individuals to quickly converge to a point.

In view of the mentioned reason, we propose the PSO algorithm for solving BPMLO by focus on decreasing number of generation for PSO_u and PSO_l , so reducing the computation time PSO_u and PSO_l respectively are PSO algorithms for upper and lower level bilevel programming. First, the algorithm initializes a particle swarm in the dominant vectors space. Then, the PSO directs the sight of swarm in the dominant vectors space together with each objective function in multi objective optimization problem, which causes the swarm to fall into the Pareto Optimal Set of the lower level program. As mentioned in section 2, PSO, inspired by the swarming behavior of animals as bird coking, was introduced by Kennedy and Eberhart (1995).

A particle swarm is a population of particles, where each particle is a moving object that is through the search space and is attracted to previously visited locations with high fitness. In contrast to the individuals in evolutionary computation, particles neither reproduce nor get replaced by other particles. Suppose particle swarm include N particles, $X_i = (x_{i,1}, \dots, x_{i,n})$ and $V_i = (v_{i,1}, \dots, v_{i,n})$ denote the location and velocity of particle i respectively in n -dimensional search space, $P_i = (p_{i,1}, \dots, p_{i,n})$ denote local optimal location of each particle (lbest), $P_{gl} = (p_{g,1}, \dots, p_{g,n})$ denote global optimal location of all particles (gbest).

Velocity update formula of each particle is:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c1rand1(p_{i,j}(t) - x_{i,j}(t)) + c2rand2(p_{g,j}(t) - x_{i,j}(t)) \quad (3.1) \quad \text{for } i = 1, \dots, N \text{ and } j = 1, \dots, n.$$

Where, t is the number of iteration, $c1, c2$ are speeding figure, regulating the length when flying to the most particle of the whole swarm and to the most optimist individual particle, w is inertia weight, $rand1$ and $rand2$ are two independent random number ranged from 0 to 1.

After updating particle velocity, use the following formula to update the location of each particle:

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t) \quad (3.2)$$

for $i = 1, \dots, N$ and $j = 1, \dots, n$.

Then compare new location of each particle with objective value of local optimal location, If the new location is better than the local optimal location, (lbest) is updated for the new location, else maintain the original value of (lbest) unchanged. Update (gbest) according to the new global optimum solution of particle swarm, and continue the evolution of next generation. After a certain number of evolutionary

computations by PSO, the particle swarm converges to the global optimum solution, so obtain a global optimal objective value.

The Algorithm

Step 0: Set up parameters, including Nmaxl the number of iterations the algorithm PSOl, Nmaxu the number of iterations the algorithm PSOu, the number of particles (Nmax), the number of maximum generations (Tsize) inertial weight(w), two acceleration coefficient (c1 and c2), Two random variables, rand1 and rand2, are in the interval [0; 1].

Step1: Generate upper level's variables, \widetilde{x}_1 , randomly.

Step 2: Solve the lower level problem:

Substep 2.1: Generate lower level variables, \widetilde{x}_2 , randomly.

Substep 2.2: Set n: = 1.

Substep 2.3: Solve the lower level problem with given \widetilde{x}_1 from step 1.

In order to check if $(\widetilde{x}_1, \widetilde{x}_2) \in IR$ or not, we use the approach by Benson [4] and check if the optimal objective value of following problem is zero:

$$\begin{aligned}
 & \max \quad \sum_{i=1}^k z_i \\
 & \text{s.t} \quad d_i x_2 + z_i = d_i \widetilde{x}_2, \quad i = 1, \dots, k \\
 & A_2^2 x_2 \leq b^2 - A_1^2 \widetilde{x}_1 \\
 & x_2 \geq 0 \\
 & z_1, \dots, z_k \geq 0
 \end{aligned} \tag{3.3}$$

Substep 2.4: Use PSO_l for improving the variable x_2 .

Substep 2.5: Check if n < Nmaxl go to 2.6, otherwise go to 2.7.

Substep 2.6: Set n: = n + 1 and go to 2.3.

Substep 2.7: Set x_2^* as the optimal solution of problem (3.3) and go to step3.

Step 3: Solve the following problem:

$$\begin{aligned}
 & \min_{x_1} f(x_1, x_2) \\
 & \text{s.t} \quad A_1^1 x_1 \leq b^1 \\
 & A_1^2 x_1 + A_2^2 x_2^* \leq b^2 \\
 & x_1 \geq 0
 \end{aligned} \tag{3.4}$$

Substep 3.1: Generate upper level variable x_{li} , randomly.

Substep 3.2: Set $n = 1$.

Substep 3.3: Solve the upper level problem with give nx_2^* from step 2.

Substep 3.4: Improve the variables with PSOu.

Substep 3.5: If $n < N_{max}$ go to 3.6, otherwise go to 3.7.

Substep 3.6: Set $n = n + 1$ and go to 3.3.

Substep 3.7: Set x_1^* as the optimal solution of problem (3.5).

Step 4: (x_1^*, x_2^*) can be considered as optimal solution for BPMLO.

4. Numerical Examples

In this section, two examples will be considered to illustrate the efficiency of the proposed algorithm. All results presented in this paper have been obtained on a personal computer with a Pentium (4)2.53 GHz processor. The codes have been written in Matlab 7.12. The PSO parameters for these examples are set as follows:

The swarm size are set to 25, The number of maximum generations, T size is set to 50, acceleration coefficient $c1 = \chi * \phi1$, $c2 = \chi * \phi2$, inertia weight $W = \chi$, where $\phi1 = 2:05$, $\phi2 = 2:05$, $\chi = \phi1 + \phi2$, $\chi = 2 / (\phi1 - 2 + \sqrt{\phi2 - 4 * \phi1})$.

Examples 4.1 consider the following problem:

$$\begin{aligned} \min_{x \geq 0} \quad & F(x, y) = x - 4y \\ \min_{y \geq 0} \quad & (y, 2y) \\ & x - y \leq -3 \\ & -2x + 4y \leq 0 \\ & 2x + y \leq 12 \\ & -3x + 2y \leq -4 \end{aligned}$$

This example is taken from [2]. By using proposed algorithm the optimal solution is achieved in fewer than 4 seconds. Also, we observed that increasing the number of iterations and especially increasing the number of population size does not have a large impact on time. Figure 1 shows the value of F with generation. As showed in Figure 1, algorithm converges to exact solution relatively fast.

For this problem, we have $(x^*, y^*) = (2, 1)$ and then $F^*(x, y) = -2$.

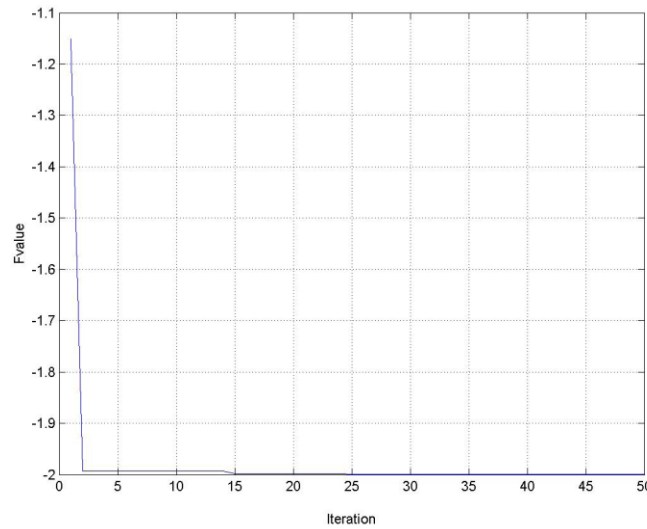


Figure 1: Value of F with generation of problem 4.1

From figure 1 it is obvious that the obtained solution by the proposed PSO algorithm is exactly the same answer in [2] which obtained in shorter time.

Example 4.2 Consider the following problem:

$$\begin{aligned}
 &\min_{x \geq 0} F(x, y, z) = x + 2y + z \\
 &\quad \min(y - 2z, -y + 2z) \\
 &\quad x + y \geq 1 \\
 &\quad x + y \leq 3 \\
 &\quad x + y + 2z \leq 5 \\
 &\quad y \geq 0, z \geq 0
 \end{aligned}$$

This problem is taken from [6]. In this problem, we have increased the dimension of the variable vector by adding a variable. Both points (1, 0, 2) and (3, 0, 0) are optimal solutions of the BPML0 and $F(1, 0, 2) = (3, 0, 0) = 3$. As before noted, an extreme point of the constraint region solves the problem. Figure 2 shows the value of F with generation. In this problem, algorithm converges to exact solution in initial iteration too.

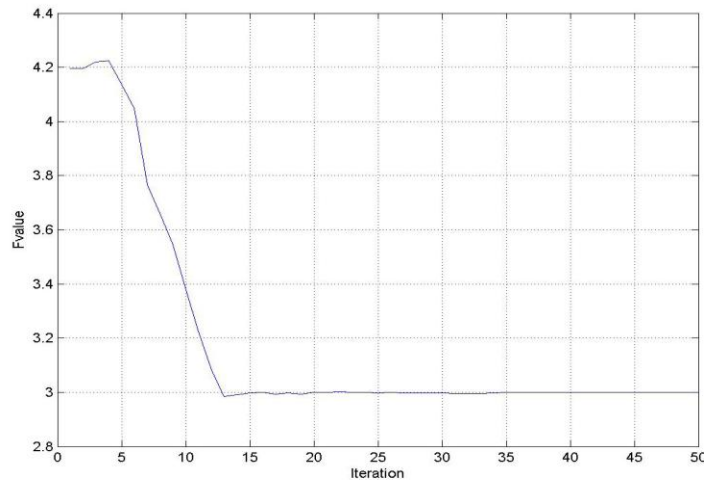


Figure 2: Value of F with generation of problem 4.2

From Figure 2 it is obvious that the obtained solution by the proposed PSO algorithm is exactly the same answer in [6].

5. Conclusion

In this paper we have interested in general bilevel problems with multiple objectives at the lower level, when all objective functions are linear and constraints at both levels define polyhedral. Meta heuristic methods are presented for reducing computational complexity. These algorithms are efficient to solve BPMLO, which is an NP-hard problem. We apply PSO algorithm to solving BPMLO. The experimental results illustrate that, obtained solutions by this algorithm are very close to the theoretical solution with a shorter time.

6. Acknowledgements

This research was made possible with help of office of vice-chancellor for researched technology, Islamic Azad University of Abadan branch. We knowledge their support and we are grateful for the funds provided by a research grant in 2012-2013.

References

- [1] M. A. Abo-Sinna, I. A. Baky, Interactive balance space approach for solving multi- level multiobjective Programming problems, Information Sciences, vol. 177, no. 16, pp. 3397-3410(2007).
- [2] G. Anandalingam, T. Fries, Hierarchical Optimization: an introduction, Annals of Operations Research 34, pp. 1-11(1992).
- [3] Z. Ankhili , A. Mansouri, An exact penalty on bilevel programs with linear vector optimization lower level European Journal of Operational Research 197, pp.36-41(2009).

- [4] J. Bard, Practical Bilevel Optimization, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, London, (1998).
- [5] H. Bonnel, J. Morgan, Semi vectorial bilevel optimization problem: penalty approach Journal of Optimization Theory and Applications 131(3), pp.65-82(2006).
- [6] Hl. Calvete and C. Gale, on linear bilevel problems with multiple objective at the lower level Journal of Omega 39, pp. 33-40(2011).
- [7] Hl. Calvete, C. Gale and P.M. Mateo, A new approach for solving linear bilevel problems using genetic algorithms European Journal of operational Research 188, 14-28(2008).
- [8] M. H. Farahi, E. Ansari, A new approach to solve Multi-objective linear bilevel programming problems, Journal of mathematics and computer Science, Volume 1, Issue 4, p 238 - 438(2010).
- [9] J. Fliege, L. N. Vicente, Multicriteria Approach to Bilevel Optimization, Journals of Optimization theory and application, Vol. 131, No. 2, p.209-225(2006).
- [10] D. Kalyanmoy and A. Sinha, Solving Bilevel Multiobjective Optimization Problems Using Evolutionary Algorithms KanGAL Report Number (2008).
- [11] R. Khalesi, H. Maleki, A new method for solving fuzzy MCDM problems, Journal of mathematics and computer Science, Volume 1, Issue 4, P.238-438(2010)..
- [12] X. Li, P. Tian, X. Min, A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems, Lecture Notes in Computer Science 4029 1169-1178(2006).
- [13] I. Nishizaki, M. Sakawa, Stackelberg solutions to multiobjective two level linear programming problems, Journal of Optimization Theory and Applications, vol. 103, no. 1pp. 161-182(1999).
- [14] M. Rostami, M. Kianpour, E. bashardoust, A numerical algorithm for solving nonlinear fuzzy differential equations, Journal of mathematics and computer Science, vol. 177, no. 16, pp. 3397-3410(2007).
- [15] Y. Shi, R.C. Eberhart, A Modified Particle Swarm Optimizer, In IEEE International Conference of Evolutionary Computation, (1998).
- [16] X. Shi, H. Xia, Interactive bilevel multiobjective decision making, Journal of the Operational Research Society, vol. 48, pp. 943-949(1997).
- [17] X. Shi, H. Xia, Model and interactive algorithm of bilevel multiobjective decision making with multiple interconnected decision makers, Journal of MultiCriteria Decision Analysis, vol. 10, pp. 27-34(2001).
- [18] Y. Zheng, Z. Wan, G. Wang, A fuzzy interactive method for a class of bilevel multi- objective programming problem, Expert Systems with Applications, vol. 38, pp. 10384-10388(2011).