

A GENETIC ALGORITHM FOR SOLVING LINEAR MULTIPLICATIVE BILEVEL PROBLEMS

H. Sadeghi¹, K. Lotfipour²

^{1,2}Department of Mathematics
Shahid Chamran University of Ahwaz
Ahwaz, IRAN

Abstract: Bilevel programming problem is a two-stage optimization problem where the constraint region of the first level problem is implicitly determined by the second level optimization problem. In this paper, we propose a genetic algorithm for solving the class of bilevel problems in which both level objective functions are linear multiplicative and the common constraint region is a bounded polyhedron. The algorithm associates chromosomes with extreme points of the polyhedron and searches for a feasible solution close to the optimal solution by introducing special crossover and mutation procedures. We report the results of numerical experiments which demonstrate the efficiency of the proposed algorithm.

AMS Subject Classification: 26A48, 52A07, 90C, 49M

Key Words: bilevel programming, genetic algorithm, extreme point, multiplicative programming

1. Introduction

Bilevel programming problem consist of two levels, namely the first level and the second level. The first level decision maker is called the leader and the second level called follower, each independently controlling part of the variables and having his own objective function. Bilevel programming structure is used for central economic planning at the regional or national level for modeling problems concerning organizational design [10], traffic assignment [7], etc. General bilevel problems can be

formulated as

$$\begin{aligned} & \max_{(x_1, x_2) \in S} f_1(x_1, x_2), \\ \text{s.t. } & x_2 \in \arg \max_{\nu \in S(x_1)} f_2(x_1, \nu), \end{aligned} \quad (1.1)$$

where $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ are the variables controlled by the first level and the second level decision makers, respectively; $f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, $n = n_1 + n_2$; $S \subset \mathbb{R}^n$ defines the common constraint region and $S(x_1) = \{x_2 \in \mathbb{R}^{n_2} : (x_1, x_2) \in S\}$. Bilevel programming are nonconvex and quite difficult to deal with [1, 9]. Notice that, these problems unlike general mathematical problems, may not possess a solution even when f_1 and f_2 (objective functions of both level) are continuous and S (common constraint region) is compact. In the recent years some surveys and bibliographic reviews appeared [6, 8] where history, applications, algorithms and almost all relevant references can be found. Bard [1] and Dempe [9] are good general references on this topic. Calvete and Gale [4] consider the case that the first level objective function is linear multiplicative and the second level is linear and the common constraint region is a polyhedron. In this paper, we focus on the class of bilevel problems in which the both level objective functions are linear multiplicative and the common constraint region S is a bounded polyhedron (LMB problem). Taking into account the existence of an extreme point of the polyhedron S which solves LMB problem [3], the aim of this paper is to combine classical enumeration techniques which search for extreme points of S with genetic algorithm. The paper organized as follows. After defining the problem in Section 2, Section 3 proceeds by developing the genetic algorithm for solving LMB problem. In Section 4 the numerical experiments are presented.

2. Description of the Problem

Here, we consider the linear multiplicative bilevel programming problem (LMB) formulated as follow:

$$\begin{aligned} & \max_{x_1} f_1(x) = (\alpha_1 + C_1^T X)(\beta_1 + D_1^T X) = (\alpha_1 + c_{11}x_1 + c_{12}x_2)(\beta_1 + d_{11}x_1 + d_{12}x_2), \\ & \max_{x_2} f_2(x) = (\alpha_2 + C_2^T X)(\beta_2 + D_2^T X) = (\alpha_2 + c_{21}x_1 + c_{22}x_2)(\beta_2 + d_{21}x_1 + d_{22}x_2), \\ \text{s.t. } & \begin{cases} A_1x_1 + A_2x_2 \leq b, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned} \quad (2.1)$$

Here $A_1 : m \times n_1$; $A_2 : m \times n_2$ and $c_{i1}, c_{i2}, d_{i1}, d_{i2}$ for $(i=1,2)$ and b are vectors of conformal dimension; α_i and β_i are scalars, and $(\alpha_i + c_{i1}x_1 + c_{i2}x_2 > 0)$, $(\beta_i + d_{i1}x_1 + d_{i2}x_2 > 0)$ for $i = 1, 2$ and all $(x_1, x_2) \in S$, and the polyhedron S is defined by the constraints (2.1). Under these assumptions, both objective functions are

quasiconcave [4]. We also assume that S is a nonempty and bounded polyhedron. The projection of S on \mathbb{R}^{n_1} is denoted by

$$S_1 = \{x_1 \in \mathbb{R}^{n_1}; (x_1, x_2) \in S\}$$

For each $\tilde{x}_1 \in S_1$ provided by the upper level decision maker, the feasible region of the second level decision maker is $S(\tilde{x}_1) = \{x_2 \in \mathbb{R}^{n_2} : A_2x_2 \leq b - A_1\tilde{x}_1, x_2 \geq 0\}$ and a feasible solution to the LMB problem is obtained by solving the following problem:

$$\begin{aligned} LM(\tilde{x}_1) : \quad & \max_{x_2} (\alpha_2 + c_{21}\tilde{x}_1 + c_{22}x_2)(\beta_2 + d_{21}\tilde{x}_1 + d_{22}x_2) \\ & s.t \begin{cases} A_2x_2 \leq b - A_1\tilde{x}_1 \\ x_2 \geq 0 \end{cases} \end{aligned} \quad (2.2)$$

Let $P(\tilde{x}_1)$ be the set of optimal solutions to (2.2). To ensure that the LMB problem is well posed, we assume that for every $\tilde{x}_1 \in S_1$, $P(\tilde{x}_1)$ is nonempty and singleton. The inducible region or feasible region of the first level decision maker is denoted by:

$$IR = \{(x_1, x_2) \in \mathbb{R}^n : x_1 \in S_1, x_2 \in P(x_1)\}$$

Any point of the set IR is a bilevel feasible solution. The bilevel problem formulated in (2.1) can be written as:

$$\begin{aligned} & \max_{x_1, x_2} (\alpha_1 + c_{11}x_1 + c_{12}x_2)(\beta_1 + d_{11}x_1 + d_{12}x_2) \\ & s.t (x_1, x_2) \in IR \end{aligned} \quad (2.3)$$

In the sequel, we use the following results that are proved in [3].

Lemma 2.1. *The inducible region of the quasiconcave bilevel programming problem (2.1) is a piecewise linear.*

Theorem 2.1. *There is an extreme point of the constraint region S which is an optimal solution to the quasiconcave bilevel programming problem (2.1).*

3. Genetic Algorithm

The main idea of the proposed algorithm is similar to EPHS¹ algorithm in calvete [5]. It looks for a good solution to the LMB problem by testing extreme points of the polyhedron S and discarding in each iteration the worst ones.

¹Extreme Points of Heuristic Search

3.1. Chromosome Encoding and Fitness Evaluation

In order to associate chromosomes with extreme points of the polyhedron S , we encode each chromosome as an m string of integers whose components are the indices of the basic variables. The fitness of a chromosome evaluates its quality. Let c be a chromosome and $(\widetilde{x}_1, \widetilde{x}_2)$ be the corresponding extreme point. The fitness value of c is defined as follow:

$$f(c) = (\delta_{IR}, (\alpha_1 + c_{11}\widetilde{x}_1 + c_{12}\widetilde{x}_2)(\beta_1 + d_{11}\widetilde{x}_2 + d_{12}\widetilde{x}_2))$$

where

$$\delta_{IR} = \begin{cases} 1 & \text{if } (\widetilde{x}_1, \widetilde{x}_2) \in IR \\ 0 & \text{otherwise} \end{cases}$$

The better fitness values are selected according to the lexicographic order. In order to compute the fitness value of chromosome c corresponding to the extreme point $(\widetilde{x}_1, \widetilde{x}_2)$, we must examine if $(\widetilde{x}_1, \widetilde{x}_2) \in IR$ or not. For this end, we check that if \widetilde{x}_2 is an optimal solution to the lower level problem defined as:

$$\begin{aligned} \max_{x_2} \quad & f_2(x) = (\alpha_2 + c_{21}\widetilde{x}_1 + c_{22}x_2)(\beta_2 + d_{21}\widetilde{x}_1 + d_{22}x_2) \\ \text{s.t.} \quad & \begin{cases} A_2x_2 \leq b - A_1\widetilde{x}_1 \\ x_2 \geq 0 \end{cases} \end{aligned} \quad (3.1)$$

we recall that K.K.T conditions are necessary and sufficient conditions for optimality of the linear multiplicative problem [2]. We apply these conditions for (3.1). So, in order to check if $(\widetilde{x}_1, \widetilde{x}_2) \in IR$, we need to check if the following linear system is feasible by applying the first phase of the two- phase method of linear programming.

$$\begin{cases} ((\beta_2 + d_{21}\widetilde{x}_1 + d_{22}\widetilde{x}_2)c_{22}^t + (\alpha_2 + c_{21}\widetilde{x}_1 + c_{22}\widetilde{x}_2)d_{22}^t) + A_2^t u - v = 0 \\ (b - A_1\widetilde{x}_1 - A_2\widetilde{x}_2)^t u = 0 \\ \widetilde{x}_2^t u = 0 \\ u \geq 0, v \geq 0 \\ \text{where } u \in \mathbb{R}^m, v \in \mathbb{R}^{n_2} \end{cases} \quad (3.2)$$

3.2. Initial Population

To create an initial population, we consider chromosomes associated with p_s extreme points of IR . For this purpose, we first get an extreme point of S , then we solve the following linear programming problem:

$$\begin{aligned} \max_{x_1, x_2} \quad & c_1x_1 + c_2x_2 \\ \text{s.t.} \quad & \begin{cases} A_1x_1 + A_2x_2 \leq b \\ x_1, x_2 \geq 0 \end{cases} \end{aligned} \quad (3.3)$$

Here the components of vectors c_1 and c_2 are randomly generated from wide range uniform distributions. In order to obtain an extreme point of IR , the following procedure is applied. Let $(\widetilde{x}_1, \widetilde{x}_2) \in S$, an extreme point is obtained by solving (3.3) for a particular objective function. Let x_2^* be the optimal solution of $LM(\widetilde{x}_1)$ using the gradient projection method of rosen [2]. If $\widetilde{x}_2 = x_2^*$ then $(\widetilde{x}_1, \widetilde{x}_2)$ is already an extreme point of IR . Otherwise $(\widetilde{x}_1, x_2^*) \in IR$, but it isn't necessarily an extreme point. In this case we get the best extreme point on the face of the polyhedron S that contains (\widetilde{x}_1, x_2^*) with respect to the upper level objective function. We solve the following problem:

$$\begin{aligned} \max_{x_1, x_2} & (\alpha_1 + c_{11}x_1 + c_{12}x_2)(\beta_1 + d_{21}x_1 + d_{12}x_2) \\ \text{s.t.} & \begin{cases} H_i(x_1, x_2) = 0 & i \in I \\ H_i(x_1, x_2) \leq 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3.4)$$

Here $H_i(x_1, x_2) \leq 0$ for $i = 1 \dots m + n_1 + n_2$, stands for the i th constraint defining S and I is the set of indices of constraints that are binding at (\widetilde{x}_1, x_2^*) . The optimal solution of problem (3.4) will be an extreme point of IR [3]. The set IR is formed by the connected union of faces of S , therefore if a point of a face is in IR [3], then all the face is in IR .

3.3. Crossover

Firstly randomly generate the percentage of the population, say p_c ($p_c \in [0, 1]$), on which the crossover is performed. Then two chromosomes are selected randomly from the population as the parents. Offsprings are generated by the following procedure:

Shared indices are eliminated and a location is randomly selected then one by one in ascending order of indices, the variables having the right-hand indices of the second parent enter the basis associated with the first parent. In other words, we consider the tableau associated with the basis of chromosome c and select the basic variables for entering. The minimum ratio test rule of simplex algorithm is applied in order to determine the variable leaving the basis. A similar process is applied for the second parent. The new chromosomes contain the indices of the new basic variables. This kind of crossover ensure that the new chromosomes associated with extreme points in S .

3.4. Mutation

Firstly, randomly generate the percentage of population, say p_m , ($p_m \in [0, 1]$) on which the mutation is performed. Then one chromosome is selected randomly from the population. Throughout the mutation, a variable is randomly selected from the set of variables whose indices aren't in this chromosome. Then this variable enter the

basis associated with chromosome and minimum ratio test rule of simplex algorithm is applied to determine the variable leaving the basis. The new chromosome contain the indices of the new basic variables.

3.5. Termination Condition

The algorithm terminates at the maximal iteration number. The best solution, which has been recorded in all iterations in the earliest time, is reported as the solution for the (LMB) problem by the proposed genetic algorithm. To build successive generations the elitist strategy is applied, i.e. after arranging the chromosomes in descending order of fitness value, the p_s chromosomes associated with the best p_s extreme points available, are kept from one generation to the next, being p_s the population size.

The outline of the algorithm is displayed as follow:

begin

Choose the population size, p_s ,

The crossover probability, p_c ,

The mutation probability, p_m and the stopping condition.

Select a p_s - sized population Q of extreme points of S.

Identify the population P of chromosomes associated with Q.

Main step

Select chromosomes from P for crossover with probability p_c .

Perform crossover.

Merge P and the offspring associated with extreme point of S.

Select chromosomes from P for mutation with probability p_m .

Perform mutation.

Merge P and the new chromosomes.

Evaluate the fitness of chromosomes in P.

Sort out all chromosomes in P in descending order of fitness values.

Remove from P all except the best p_s chromosomes.

While the stopping condition is not verified repeat the main step.

end

4. Numerical Experiments

To demonstrate the feasibility and efficiency of the proposed algorithm, we carry out the following numerical examples. We consider $p_s = 20$, $p_c = 0.5$, $p_m = 0.4$ and the set $T = 20$, (T is the number of iterations). The numerical experiments were performed on a PC pentium 4 under *Windows XP*. All codes were written in

MATLAB 7, Release 14.

Example 4.1.

$$\begin{aligned}
 & \max_{x_1} (3 + x_1 - x_2)(70 - 5x_1 - 3x_2) \\
 & \max_{x_2} (9 + x_1 + x_2)(4 + 3x_1 + x_2) \\
 & s.t \begin{cases} -5x_1 - 3x_2 \leq -15 \\ -x_1 + 4x_2 \leq 28 \\ 2x_1 + 3x_2 \leq 32 \\ x_1 + x_2 \leq 13 \\ 2x_1 - x_2 \leq 13 \\ x_1 - 4x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases} \tag{4.1}
 \end{aligned}$$

Example 4.2.

$$\begin{aligned}
 & \max_{x_1, x_2} (2 + x_1 - x_2 + 2x_4)(9 - x_1 - 2x_3 + x_4 + 5x_5) \\
 & \max_{x_3, x_4, x_5} (1 + x_1 + x_2 + 2x_3 - x_4 + x_5)(6 + 2x_1 + x_3 + x_4 - 3x_5) \\
 & s.t \begin{cases} -x_3 + x_4 + x_5 \leq 1 \\ 2x_1 - x_3 + 2x_4 - 0.5x_5 \leq 1 \\ 2x_2 + 2x_3 - x_4 - 0.5x_5 \leq 1 \\ x_i \geq 0 \quad i = 1 \dots 5 \end{cases} \tag{4.2}
 \end{aligned}$$

No	x_1	x_2	f_1	f_2
1	8.6667	4.3333	100.2222	755.3448
2	(0, 0)	(1.5, 1.5, 1)	62.5	21

Table 1

Note. f_1 and f_2 are the objective function values of the upper level and lower level programming problem, respectively. (x_1, x_2) is the optimum of the problem.

5. Conclusions

In this paper we have developed a genetic algorithm for (LMB) problems. It searches among extreme points of the polyhedron S , which are associated with chromosomes. In fact, the strenght of the algorithm is that it can be applied to any bilevel problem having an optimal solution which is an extreme point of S , since this is the only requirement for applying of the method.

References

- [1] J.F. Bard, *Practical Bilevel Optimization. Algorithms and Applications*, Dordrecht, Kluwer Academic (1998).
- [2] M.S. Bazara, H.D. Sherali, C.M. Shetty, *Nonlinear Programming. Theory and Algorithms*, Second Edition, New York, Wiley (1993).
- [3] H.I. Calvete, C. Gale', On the quasiconcave bilevel programming problem, *Journal of Optimization Theory and Applications*, **98**, No. 3 (1998), 613-622.
- [4] H.I. Calvete, C. Gale', Bilevel multiplicative problems: A penalty approach to optimality and a cutting plane based algorithm, *Journal of Computational and Applied Mathematics*, **218** (2008), 259-269.
- [5] H.I. Calvete, C. Gale', P.M. Mateo, A genetic algorithm for solving linear fractional bilevel problems, *Annals of Operation Research*, **166** (2009), 39-56.
- [6] B. Colson, P. Marcotte, G. Savard, Bilevel programming: A survey, *4OR: O.J. Oper. Res.*, **3** (2005), 87-107.
- [7] D.P. Bersekas, E. Gafni, Projection method for variational inequalities and application to a traffic assignment problem, *Mathematical Programming Study*, **17** (1982), 139-159.
- [8] S. Dempe, Annotated bibliography on bilevel programming and mathematical problems with equilibrium constraints. *Optimization*, **52** (2003), 333-359.
- [9] S. Dempe, *Foundation of Bilevel Programming*, Dordrecht, Kluwer Academic (2002).
- [10] P.T. Harker, Alternative models of spatial competition, *Operation Research*, **34** (1986), 410-425.