Solving Method for a Class of Bilevel Linear Programming based on Genetic Algorithms*

G. Wang†, Z. Wan‡ and X. Wang§

Abstract

The paper studies and designs an genetic algorithm (GA) of the bilevel linear programming problem (BLPP) by constructing the fitness function of the upper-level programming problem based on the definition of the feasible degree. This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints. Finally, the numerical results of some examples indicate the feasibility and effectiveness of the proposed method.
Key words: bilevel linear programming, genetic algorithm, fitness function
AMS Subject Classification 65K05 90C99 93A13

1 Introduction

Bilevel linear programming problem is one of the basic types of bilevel systems. Although the objective function and the constraints of the upper-level and lower-level problems of BLPP are all linear, the BLPP is neither continuous everywhere nor convex for the objective function of the upper-level problem is decided by the solution function of the lower-level problem which, generally speaking, is neither linear nor differentiable. Bialas and Karwan proved the non-convexity of the upper-level problem with practical example [1]. Bard，Ben-Ayed and Blair proved BLPP is a NP-Hard problem [2, 3]. Besides, Hansen, Jaumard and Savard proved BLPP is a strongly NP-Hard problem [4]. Later, Vicente,Savard and Judice declared it is a NP-Hard problem to find the local optimal solution of BLPP [5].

At present, the various algorithms developed for solving the BLPP can be classified into the following categories:

1. Extreme-point search method. It mainly includes K-best algorithm from Bialas and Karwan, Wen and Bialas [1, 7, 8], Grid-search algorithm from Bard [9, 10], etc.

2. Transformation method. It mainly includes complementary-pivot algorithm from Judice and Faustino, Candler and Townsley, Onal [11, 12, 13, 14], branch-and-bound algorithm

*Supported by the National Natural Science Foundation(60274048) and the Doctoral Foundation in Ministry of Education of China(20020486035) and the Huazhong Grid Bureau of National Electric Power Company.
†Graduate Student, School of Mathematics and Statistics, Wuhan Univ., Wuhan 430072 China.
‡Corresponding author. E-mail:zpwan@public.wh.hb.cn. Professor, School of Mathematics and Statistics, Wuhan Univ., Wuhan 430072 China.
§Professor, Institute of Systems Engineering, Wuhan Univ., Wuhan,430072 China.

from Bard and Falk, Bard and Moore, Hansen, Jaumard and Savard [4, 15, 16], penalty-function approach method from White and Anandalingam, Aiyoshi and Shimizu [17, 18, 19], etc.

3. The evolutionary method. It mainly includes the GA solving the Stackelberg-Nash equilibrium [20], and the GA of Mathieu, Pittard and Anandalingam [21], etc.

The paper studies and designs the genetic algorithms of BLPP by constructing the fitness function of the upper-level programming problem based on the definition of the feasible degree. This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints. Finally, the numerical results of some examples indicate the feasibility and effectiveness of the proposed method

2 The Mathematical Model of the BLPP

Candler and Townsley proposed the common mathematical model of BLPP[13]. Later, Bard, Bard and Falk, Bialas and Karwan proposed various mathematical models of BLPP[1, 6, 7, 15, 22]. Here, we consider the following BLPP with the form:

$$\begin{aligned} (P1) max F(x,y) &= a^T x + b^T y \\ \text{where} \quad y \quad \text{solves} & \\ (P2) max f(x,y) &= c^T x + d^T y \\ \text{subject to} & \\ Ax + By &\le r \\ x, y &\ge 0 \end{aligned}$$

where $A \in R^{m\times n_1}$, $B \in R^{m\times n_2}$, $a, c, x \in R^{n_1}$, $b, d, y \in R^{n_2}$, $r \in R^m$

Once x is fixed, the term $c^T x$ in the objective function of the lower-level problem is a constant. So the objective function of the lower-level problem is simply denoted as:

$$\overline{f} = d^T y$$

Let $S = \{(x,y)|Ax + By \le r\}$ denote the constraint region of BLPP. Here, we assume S is nonempty and bounded. Let $Q(x) = \{y|By \le r - Ax, y \ge 0\}$ nonempty and bounded. Let $Y(x)$ denote the optimal solution set of the problem $\max_{y\in Q(x)} \overline{f} = d^T y$. We assume the element of the set $Y(x)$ exists and is unique, then the inducible region is: $\psi_f(S) = \{(x,y)|(x,y) \in S, y = Y(x)\}$

Hence, the problem (P1) can be changed into:

$$(P3)\qquad \begin{array}{rcl} maxF(x,y) & = & c^T x + d^T y \\ \text{subject to} & & \\ Ax + By & \leq & r \\ y = Y(x) & & \end{array}$$

Then, if the point (x, y) is the solution of the following problem

$$\begin{array}{rcl} maxF(x,y) & = & c^T x + d^T y \\ \text{subject to} & & \\ Ax + By & \leq & r \end{array}$$

and $y = Y(x)$, then (x, y) is the solution of the problem $(P1)$.

Definition 1: The point (x, y) is feasible if $(x, y) \in \psi_f(S)$

Definition 2: The feasible point $(x^*, y^*) \in \psi_f(S)$ is the optimal solution of the BLPP (solution for short) if $F(x^*, y^*) \geq F(x, y)$ for each point $(x, y) \in \psi_f(S)$.

The paper will discuss the numerical method of BLPP under the definition.

3 Design of the GA for BLPP

It is not easy to know the upper-level objective function of BLPP has no explicit formulation, since it is compounded by the lower-level solution function which has no explicit formulation. Thus, it is hard to express the definition of the derivation of the function in common sense. And it is difficult to discuss the conditions and the algorithms of the optimal solution with the definition. We concerned the GA is a numerical algorithm compatible for the optimization problem since it has no special requirements for the differentiability of the function. Hence the paper solves BLPP by GA.

The basic idea solving BLPP by GA is: firstly, choose the initial population satisfying the constraints, then the lower-level decision maker makes the corresponding optimal reaction and evaluate the individuals according to the fitness function constructed by the feasible degree, until the optimal solution is searched by the genetic operation over and over.

3.1 Coding and Constraints

At present, the coding often used are binary vector coding and floating vector coding. But the latter is more near the space of the problem compared with the former and experiments show the latter converges faster and has higher computing precision[23] . The paper adopts the floating vector coding. Hence the individual is expressed by: $v_k = (v_{k1}, v_{k2}, \ldots, v_{km})$.

The individuals of the initial population are generally randomly generated in GA, which tends to generate off-springs who are not in the constraint region. Hence, we must deal with them.

Here, we deal with the constraints as follows: generate a group of individuals randomly, then retain the individuals satisfying the constraints $Ax + By \leq r$ as the initial population and drop out the ones not satisfying the constraints. The individuals generated by this way all satisfy the constraints. And, the off-springs satisfy the constraints by corresponding crossover and mutation operators.

3.2 Design of the Fitness Function

To solve the problem $(P3)$ by GA, the definition of the feasible degree is firstly introduced and the fitness function is constructed to solve the problem by GA. Let d denote the large enough penalty interval of the feasible region for each$(x, y) \in S$:

Definition 3: Let $\Theta \in [0,1]$ denotes the feasible degree of satisfying the feasible region, and describe it by the following function:

$$\Theta = \begin{cases} 1, & \text{if } \ \|y - Y(x)\| = 0 \\ 1 - \|\frac{y-Y(x)}{d}\|, & \text{if } \ 0 < \|y - Y(x)\| \leq d \\ 0, & \text{if } \ \|y - Y(x)\| > d \end{cases}$$

where $\|.\|$ denotes the norm.

Further,the fitness function of the GA can be stated as:

$$eval(v_k) = (F(x, y) - F_{min}) * \Theta$$

where F_{min} is the minimal value of $F(x, y)$ on S.

3.3 Genetic Operators

The crossover operator is one of the important genetic operators. In the optimization problem with continuous variable, many crossover operators appeared, such as[23]: simple crossover, heuristic crossover and arithmetical crossover. Among them, arithmetical crossover has the most popular application. The paper uses arithmetical crossover which can ensure the off-springs are still in the constraint region and moreover the system is more stable and the variance of the best solution is smaller. The arithmetical crossover can generate two off-springs which are totally linear combined by the father individuals. If v_1 and v_2 crossover, then the final off-springs are:

$$v_1^{'} = \alpha * v_1 + (1 - \alpha) * v_2$$
$$v_2^{'} = \alpha * v_2 + (1 - \alpha) * v_1$$

where $\alpha \in [0,1]$ is a random number. The arithmetical crossover can ensure closure (that is,$v_1, v_2 \in S$)

The mutation operator is another important genetic operator in GA. Many mutation operators appeared such as[23]: uniform mutation, non-uniform mutation and boundary mutation. We adopt the boundary mutation, which is constructed for the problem whose optimal solution is at or near the bound of the constraint search space. And for the problem with constraints, it is proved to be very useful. If the individual v_k mutates, then

$$v_k^{'} = (v_{k1}^{'}, v_{k2}^{'}, \ldots, v_{km}^{'})$$

where $v_{ki}^{'}$ is either left($v_{ki}^{'}$) or right($v_{ki}^{'}$) with same probability(where, left($v_{ki}^{'}$), right($v_{ki}^{'}$) denote the left, right bound of $v_{ki}^{'}$, respectively)

Selection abides by the principle: the efficient ones will prosper and the inefficient will be eliminated, searching for the best in the population. Consequently the number of the superior individuals increases gradually and the evolutionary course goes along the more optimization. There are many selection operators. We adopt roulette wheel selection since it is the simplest selection.

3.4 Termination criterions

The judgment of the termination is used to decide when to stop computing and return the result. We adopt the maximal iteration number as the judgment of the termination. The algorithm process using the GA is as follows:

Step 1 Initialization. give the population scale M, crossover probability P_c, mutation probability P_m, the maximal iteration generation $MAXGEN$, and let the generation t=0;

Step 2 Initialization of the initial population. M individuals are randomly generated in S, making up of the initial population.

Step 3 Computation of the fitness function. Evaluate the fitness value of the population according to the formula (1) .

Step 4 Generate the next generation by genetic operators. Select the individual by roulette wheel selection, crossover according to the formula (2) and mutate according to the formula (3) to generate the next generation.

Step 5 Judge the condition of the termination. When t is larger than the maximal iteration number, stop the GA and output the optimal solution. Otherwise, let $t = t + 1$, turn to Step 3

4 The Numerical Results

We proposed the following example to show the efficiency of the GA method solving the BLPP:

Example 1[24]

$$\begin{aligned}\max_{x\geq 0} F(x,y) &= x+3y\\ \text{where} \quad y \quad &\text{solves}\\ \max_{y\geq 0} f(x,y) &= x-3y\\ \text{subject to}&\\ -x-2y &\leq -10\\ x-2y &\leq 6\\ 2x-y &\leq 21\\ x+2y &\leq 38\\ -x+2y &\leq 18\end{aligned}$$

Example 2[13]

$$\begin{aligned}\max_{x\geq 0} F(x,y) &= -8x_1-4x_2+4y_1-40y_2-4y_3\\ \text{where} \quad y \quad &\text{solves}\\ \max_{y\geq 0} f(x,y) &= x_1+2_2+y_1+y_2+2_y3\\ \text{subject to}&\\ -y_1+ \quad y_2 \quad &+y_3 \leq 1\\ 2x_1-y_1+ \quad 2y_2 \quad &+0.5y_3 \leq 1\\ 2x_2+2y_1- \quad y_2 \quad &-0.5y_3 \leq 1\end{aligned}$$

Example 3[4]

$$\begin{aligned}\max_{x\geq 0} F(x,y) &= 8x_1+4x_2-4y_1+40y_2+4y_3\\ \text{subject to}&\\ x_1+ \quad 2x_2 \quad &-y_3 \leq 1.3\\ \max_{y\geq 0} f(x,y) &= -2y_1-y_2-2y_3\\ \text{subject to}&\\ -y_1+ \quad y_2 \quad &+y_3 \leq 1\\ 4x_1-2y_1+ \quad 4y_2 \quad &+y_3 \leq 2\\ 4x_2+4y_1- \quad 2y_2 \quad &-y_3 \leq 2\end{aligned}$$

The compare of the results thorough 500 generations by the algorithms in the paper and the results in the references is as follows:

	parameters			*result in the paper*			*result in the references*		
	M	P_c	P_m	(x,y)	F	f	(x,y)	F	f
1	50	0.7	0.15	$(15.959, 10.972)$	48.875	-16.957	$(16, 11)$	49	-17
2	100	0.6	0.2	$(0, 0.899, 0, 0.6, 0.394)$	29.172	-3.186	$(0, 0.9, 0, 0.6, 0.4)$	29.2	-3.2
3	100	0.6	0.3	$(0.533, 0.8, 0, 0.197, 0.8)$	18.544	-1.797	$(0.2, 0.8, 0, 0.2, 0.8)$	18.4	-1.8

Notes: M denotes the population scale, P_c denotes crossover probability, P_m denotes mutation probability. F and f are the objective function value of the upper-level and lower-level programming problem, respectively.

From the numerical result, the results by the method in this paper accord with the results in the references. So the method is very efficient. In addition, the experiments show: the scale of the population and the rate of the mutation have some influence on the efficiency and convergence rate of the method, but crossover operator has little influence on them. For example, the rate of convergence will slow down if the size of the population is larger, and it is benefit for the rate of the convergence to choose the lager rate of mutation.

5 The Conclusion

The paper proposes the GA of the BLPP of which the optimal solution of the lower-level problem is dependent on the upper-level problem. The numerical results show the method is feasible and efficient. Compared with the traditional methods, the method has the following characters:

1. The method has no special requirement for the characters of the function and overcome the difficulty discussing the conditions and the algorithms of the optimal solution with the definition of the differentiability of the function.

2. This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints.

References

[1] W.F.Bialas and M.H.Karwan, *Two-level linear programming. Management Science.* 30, 1004-1020 (1984)

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

[3] O. Ben-Ayed and C.E.Blair, *Computational difficulties of bilevel linear programming. Operations Research.* 38, 556-560 (1990)

[4] P. Hansen, B.Jaumard and G. Savard, *New branch-and-bound rules for linear bilevel programming. SIAM Journal on Science and Statistical Computing.* 13(5), 1194-1217 (1992)

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

[6] J.F.Bard. *Practical Practical Bilevel Optimization Algorithms and Applications, Kluwer Academic Publishers.* (1998)

[7] W.F.Bialas and M.H.Karwa. *On two-level optimization. IEEE Transactions Automatic Conrrol.* 27(1), 211-214 (1982)

[8] Wen,U.P. and W.F.Bialas. *The hybrid algorithm for solving the three-level linear programming problem.Computers and Operations Research.* 13(4), 367-377 (1986)

[9] J.F.Bard. *A grid search algorithm for the linear bilevel programming problem.in* 14^{th} *Annual Meeting of the American Institute for Decision Science,San Francisco,CA.* 2, 256-258 (1982)

[10] J.F.Bard. *An efficient point algorithm for a linear two-stage optimization problem. Operations Research.* 31,670-684 (1983)

[11] J.J.Judice and A.M.Faustino. *n expetimental investigation of enumerative methods for the linear complementarity problem.Computers and Operations Research.* 15(5), 417-426 (1988)

[12] J.J.Judice and A.M.Faustino. *A sequential lcp method for bilevel linear programming.Annals of Operations Researc.* 34, 89-106 (1992)

[13] W.Candler and R.Townsley. *A linear two-level programming problem. Computers and Operations Reseatch.* 9, 59-76 (1982)

[14] H.Onal. *A modified simplex approach for solving bilevel programming problems.European Journal of Operational Research.* 67(1), 126-135 (1993)

[15] J.F.Bard and J. E. Falk. *An explicit solution to multi-level programming problem. Computers and Operations Reseatch.* 9, 77-100 (1982)

[16] J.F.Bard and J.T.Moore. *A branch and bound algorithm for the bilevel programming problem.SIAM Journey on Scientific and Statistical Computing.* 11, 281-292 (1990)

[17] D.J.White and G.Anandalingam. *A penalty function approach for solving bilevel linear programs. Journal of Global Optimization.* 3(4), 397-419 (1993)

[18] E.Aiyoshi and K.Shimizu. *A new computational method for Stackelberg and min-max problem by use of a penalty method.IEEE Transactions on Automatic Control.* 26(2),460-466 (1981)

[19] E.Aiyoshi and K.Shimizu. *A solution method for the static constrained Stackelberg problem via penalty method.IEEE Transactions on Automatic Control.* 29(12), 1111-1114 (1984)

[20] B.Liu. *Stackelberg-Nash equilibrium for multilevel programming with multiple follows using genetic algorithms. Computers Math.Applic..* 36(7),79-89. (1998)

[21] R.Mathieu,L.Pittard and G.Anandalingam. *Genetic algorithm based approach to bilevel linear programming.R.A.I.R.O.Recherche Operationelle.* 28, 1-21 (1994)

[22] J.F.Bard. *Geometric and algorithmic developments for a hierarchical planning problem. Computers and Operational Reseatch.* 19, 372-383 (1985)

[23] Z. Michalewicz. *Genetic Algorithms + Data Structures=Evolution Programs, Springer-Verlag, New York.* (1996)

[24] G.Anandalingam and D.J.White. *A solution for the linear static Stackelberg problem using penalty function. IEEE Transactions Automatic Control.* 35, 1170-1173 (1990)