A Genetic Algorithm for Solving a Class of Multi-objective Bilevel Programming Problems

Shanfeng Zhang, Mengwei Li

Department of Computer, **** University, Guangzhou, 810008, China

Keywords: multi-objective bilevel programming, genetic algorithm, interpolation.

Abstract. At present, most of the researches on bilevel programming problems are focused on single objective cases. This paper discusses a bilevel programming problem with upper level multi-objective optimization. In order to solve the problem efficiently, we present a genetic algorithm using interpolation. This method does not require solving lower optimization problem frequently. In the proposed algorithm, firstly, the interpolation functions are adopted to approximate lower level solution functions. As a result, the original problem can be approximated by a single-level multi-objective programming. In addition, the corresponding interpolation functions are updated such that these functions can approximate the lower level solution function very well. Finally, the multi-objective programming is solved for obtaining an optimal solution set of the original problem. The simulation on two examples indicates the proposed algorithm is effective and feasible.

1 Introduction

Bilevel programming problem (BLPP) is a hierarchical optimization problem, which includes an upper levels decision maker and a lower levels decision maker. Both of them have the objective functions and constraint conditions. This problem was proposed by Stackelberg[1], and known as Stackelberg problem.

2 The lower solution based on interpolation

Interpolation function

For each feasible point (x, y), it is necessary to solve lower one level problem. For large-scale problems the amount of calculation increased significantly.

For each x, the lower level optimal solution $y(x)$ is unique. The problem can be transformed into a multi-objective single-level optimization problem :

$$\begin{cases} \min\limits_{x\in X} & (F_1(x, y(x)), F_2(x, y(x)), \cdots, F_m(x, y(x))) \\ s.t. & G(x, y(x)) \le 0 \end{cases} \tag{3}$$

Where, $y(x) = (y^1(x), y^2(x), \cdots, y^q(x))^T$, y^i can be seen as a function of upper variables. However, it is difficult to obtain $y^i = y^i(x)$ .We use interpolation function $\phi(x)$ to approximate the optimal solution function.

Hence, (3) can be transformed into (4)

$$\begin{cases} \min\limits_{x\in X} & (F_1(x, \phi(x)), F_2(x, \phi(x)), \cdots, F_m(x, \phi(x))) \\ s.t. & G(x, \phi(x)) \le 0 \end{cases} \tag{4}$$

Given N_1 points $x_i = (x_i^1, x_i^2, \cdots, x_i^p)$, $i = 1,2,\cdots, N_1$, these points are fixed in the lower level functions, and the lower level problems are solved to obtain the lower level optimal solution via a genetic algorithm. As a result, N_1 interpolation nodes are obtained as follows

$$(x_i, y_i) = (x_i^1, x_i^2, \cdots, x_i^p, y_i^1, y_i^2, \cdots, y_i^q) \text{ , } i = 1,2,\cdots, N_1$$

Then, we use a cubic spline interpolation function in the MATLAB toolbox to get an approximate function that is $y \approx \phi(x)$ .In the proposed algorithm, we update the interpolation nodes and interpolation functions. This process makes the interpolation function approximation get better and better.

Fast non-dominated sorting.

These individuals in the population are sorted based on non-domination, the fast non-dominated sorting algorithm is adopted as follows [7]:

i) For each individual i, there are two parameters(n_i and S_i)have been defined. n_i represents the number of individuals that dominate i in the population, whereas S_i represents the set of individuals that are dominated by i.

ii) We find out all individuals which $n_i = 0$,and add them to the set F_1 which stores the individual rank is one, i.e. $i_{rank} = 1$.

iii) We consider each individual j in F_1, check the set S_j which stores the individuals dominated by individual j .For each individual k in S_j ,if $n_k - 1 = 0$, they will be stored in another collection F_2 .Set rank of individual k to second

iv) This process is repeated until all individuals get their rank values.

3 Genetic algorithm

Genetic algorithm is widely used in the multi-objective optimization problems with global search capability and robustness[7]. In order to solve the problem (3), we encode the upper variable values using real coding scheme, and give a fitness function based on non-dominated solutions sorting method and crowding distance, it can distinguish different individuals effectively. Our algorithm is developed:

Step1: (Interpolation function) Get N_1 individuals x_i in X randomly. By using genetic algorithms, we obtained the corresponding lower level optimal solution y_i and get the interpolation function $\phi(x)$, which (x_i, y_i) are nodes.

Step2: (Initial population)Take N points randomly, and substitute these points to interpolation function $\phi(x)$ to obtain the initial population with a population size N

Step3: Arithmetic crossover operator.[8]

Step4: Non-uniform mutation operator.[8]

Step5: (Select)Set $O = pop(k) \cup O_1 \cup O_2$, execute the fast non-dominated sorting, determine their ordinal values, and calculate the crowding distance of the individuals with the same ordinal value. Define the relationship $\prec$:For two different individuals i and j, if $i_{rank} < j_{rank}$ or $i_{rank} = j_{rank}$, $i_d > j_d$ we call $i \prec j$. Choose N individuals from O as $pop(k+1)$.

Step6: (Update interpolation function) The individuals with $i_{rank} = 1$ are selected from O, and we suppose there are m points. For these m points, the lower level problems are solved, and other m nodes are gotten. These points are used to update the interpolation function. In order to reduce the amount of computation of obtaining the lower level solutions, we design a multi-criteria evolutionary scheme. Firstly, for each point x, generate t points according to Gaussian distribution. Hence, we get a population with population size $t \cdot m$. m fitness functions are obtained by m upper level variable values in the evaluation and selection process, which makes m runs of genetic algorithm are finished in one execution.

Step7: (Termination condition) If the algorithm reach the maximum generation T, then stops, and output the best non-dominated individuals; otherwise let $k = k + 1$, turn to Step3.

4 Computational examples and analysis

In order to illustrate the feasibility and effectiveness of the algorithm, we construct two examples according to examples in literature [9, 10]. We solve them by two different approaches. The first is the approach proposed in this paper, and the second approach is the same as the first one except using the MATLAB toolbox function to solve the lower level problem. The parameters are set as follows: the population size is 100, the maximum generation is 50, and the crossover and mutation probability is 0.8 and 0.1.

Example 1：

$$\begin{cases} \min & (-(x_1+2x_2),-(3x_1+x_2)) \\ s.t. & x_1+x_2\le 3 \\ \min & -(3y_1+4y_2) \\ s.t. & -x_1+y_1+y_2\le 6 \\ & x_1+x_2+y_2\le 8 \\ & -x_2+y_1\le 3 \\ & x_1,x_2,y_1,y_2\ge 0 \end{cases}$$

Example 2：

$$\begin{cases} \min & (-2x_1-3y_1-y_2,-x_2+y_1-3y_2) \\ s.t. & 0\le x_1\le 10, 0\le x_2\le 5 \\ \min & -4y_1+7y_2 \\ s.t. & y_1-y_2-x_1\le 3 \\ & y_1-y_2+x_2\le 4 \\ & 0\le y_1\le 15, 0\le y_2\le 20 \end{cases}$$

The pareto frontier of the two examples are shown as follows:

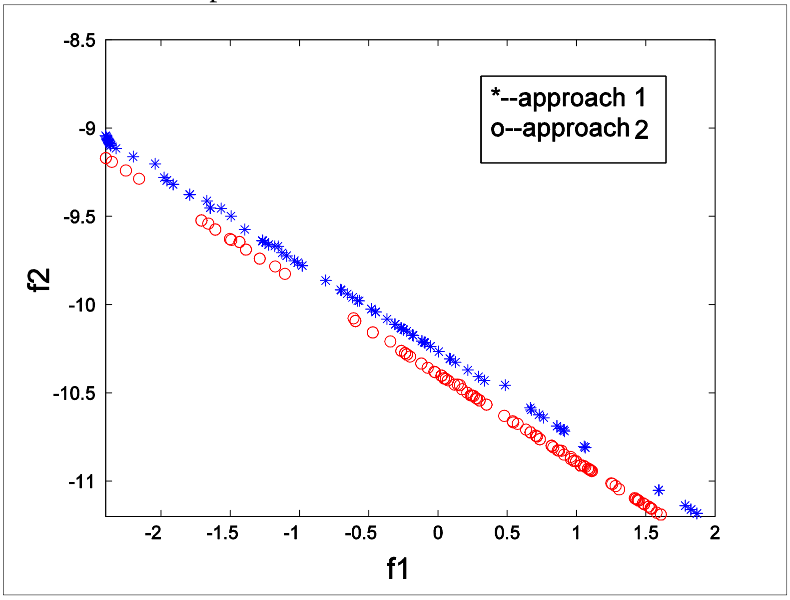

Fig.1: Pareto front of example 1

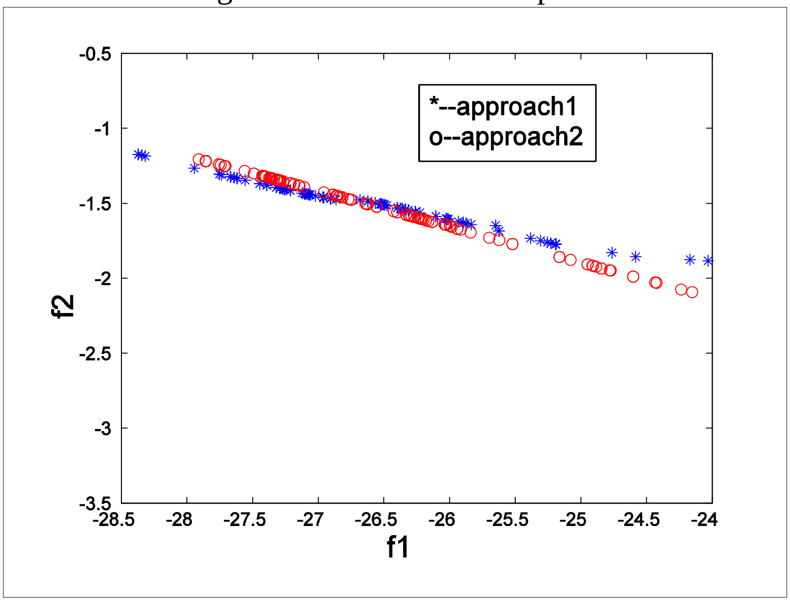

Fig.2: Pareto front of example 2

Table 1: Cpu time of the two approaches for each problem

No.	Approach1 for problem1	Approach2 for problem1	Approach1 for problem2	Approach2 for problem2
1	10.3594	64.4531	11.0781	64.0937
2	10.1719	65.6406	11.5625	59.8445
3	10.3438	63.3750	11.7188	61.9219
4	11.2188	66.0469	11.3281	58.6250
5	103594	65.2656	11.3125	59.4063

As can be seen from Fig.1,Fig.2 and Table 1, The optimal results are very close. But the Cpu time of Approach 1 is far less than that of Approach 2. It can be seen that the proposed algorithm is feasible and effient.

5 Conclusions

In the proposed algorithm, the multi-objective bilevel programming problem is transformed into a single-level problem by using interpolation functions of the lower level solutions. The process avoids solving the lower level problems frequently, and reduces the computational cost. The major advantage of this algorithm is that it can solve some complex issues, in which the lower level problems are non-convex and non-differentiable. Hence, it can be used to deal with hard multi-objective bilevel programming problems.

6 Acknowledgements

The research work was supported by National Natural Science Foundation of China under Grant No. 61065009 and Natural Science Foundation of Qinghai Provincial under Grant No. 2011-z-756.

References

[1] Stackelberg H. *The Theory of the Market Economy*. Oxford: Oxford University Press, 1952.

[2] Miller,T., Friesz,T. & Robin,R., Heuristic algorithms for delivered pric spatially competitive network facility location problem. Ann. Oper. Res., 34,pp.177-202,1992.

[3] Pieume, C.O, Fotso L.P. & P. Siarry. A method for solving bilevel linear progremming problem. *Journal of Information and Optimization Science,* 29(2), pp. 335-358,2008.

[4] Aryanezhad,M.B. & Roghanian, E.A., Bilevel linear multi-objective decision making model with interval coefficents for suppy chain coordination.*Iust Internation Journal of Engineering Science*, 19(1-2), pp.67-74,2008,.

[5] Cheng-Min Feng & Chieh-chao Wen, A fuzzy bi-level and multi-objective model to control traffic flow into the disaster area post earthquake. *Journal of the Eastern Asia Society for Transportation Studies*,6,pp.4253-4268,2005.

[6] Li He-Cheng, Wang Yu-Ping, An interpolation based genetic algorithm for sloving nonlinear bilevel programming problems.*Chinese Journal of Computers*, 31(6), pp.910-918, June 2008.

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal & Meyarivan, T.A., Fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comp., 6(2), pp.182 – 197,2002.

[8] Wang Yu-Ping, *The Theory and Method of Evolutionary Computation,* Beijing: Science Press,2011.

[9] Farahi, M. H. & Ansari, E. A new approach to solve muti-objective linear bilevel programming problems. *Journal of Mathematics and Computer Science,* 1(4), pp.313-320,2010.

[10]Eman,O.E., Interactivebi-level multi-objective integer non-linear programming problem. *Applied Mathematical Sciences*,5(65),3221-3232,2011.