Applied Mathematics & Information Sciences 5 (2) (2011), 139S-146S
– An International Journal

An Evolutionary Algorithm with Local Search for Convex Quadratic Bilevel Programming Problems

Hecheng Li[1] and Yuping Wang[2]
[1] Department of Mathematics and Information Science, Key Laboratory of Tibetan Information Processing of Ministry of Education, Qinghai Normal University, Xining 810008, China
[2] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
Email Address: ywang@xidian.edu.cn

Received June 22, 2010; Revised March 2, 2011

We present an evolutionary algorithm based on a local search scheme for convex quadratic bilevel programming problems. At first, based on Karush-Kuhn-Tucher conditions, the follower's problem is replaced by its linear complementarity systems, and the bases of the linear systems are encoded as individuals of populations. For each fixed individual *l*, we can obtain the follower's optimal solution *y*(*x*). In addition, we replace the follower vector *y* in the leader's problem by the solution function *y*(*x*), and then the leader's problem is converted to a convex quadratic problem which only involves *x*. At last, we solve the problem using a classical local search technique and obtain an optimal objective value *F*, which is taken as the fitness of *l*. The distinct characteristic of the algorithm is that the bases of the follower's complementarity systems are searched instead of the variable values, which makes the search space become finite. In order to illustrate the efficiency of the proposed algorithm, 10 test problems selected from literature are solved, and the results show that the proposed algorithm is efficient and robust.

Keywords: Convex quadratic bilevel programming problems, evolutionary algorithm, local search, optimal solutions.

1 Introduction

Bilevel programming problem (BLPP) involves two players at different levels, the leader and the follower. Both the leader and the follower have own decision variables and objectives, and make an attempt to optimize their own objectives in sequence. The general bilevel programming problem can be formulated as follows

$$\begin{cases} \min\limits_{x \in X} F(x, y) \\ s.t.\ G(x, y) \le 0 \\ \min\limits_{y \in Y} f(x, y) \\ s.t.\ g(x, y) \le 0 \end{cases} \tag{1.1}$$

Where *x* and *y* are called leader's and follower's vectors, whereas *F* and *f* are called leader's and follower's objective functions, respectively. In BLPP (1.1), the leader moves first by choosing a vector *x* to optimize *F*. For each fixed *x*, the follower optimizes his/her objective function *f* by selecting a vector *y* which is an optimal solution to the follower programming.

BLPP exists in the area of engineering and economic management extensively, it is important to design effective and efficient algorithms to solve kinds of BLPPs. In spite of the fact that a variety of exact algorithmic approaches have been developed for dealing with linear and nonlinear BLPPs[1-2,4-10], due to the nonconvexity of BLPPs, the most of existing algorithms are very time consuming and can't deal with large-scale problems in a reasonable time of computation. In order to overcome these shortcomings, the paper is devoted to designing an evolutionary algorithm (EA) which can reduce the computational complexity caused by solving follower's problems. We consider convex quadratic BLPPs, that is, the objectives are convex quadratic functions, whereas the constraints are linear with respect to (*x*, *y*). In order to solve this class of nonlinear bilevel programming problems efficiently, some special techniques are used in the proposed algorithm. At first, based on Karush-Kuhn-Tucher (K-K-T) conditions, the follower's problem is replaced by its linear complementarity systems, and the bases of the linear systems are encoded as individuals in populations. For each fixed individual *l* which corresponds to a base *B*, we can obtain the follower's optimal solution *y*(*x*) when *B* is regarded as an optimal base for some *x*. In addition, we replace all follower variables *y* in the leader's problem by the solution function *y*(*x*) and take $y \ge 0$ into account; as a result, the leader's problem is converted to a convex quadratic problem only involving *x*. At last, we solve the problem using a classical search technique and obtain an optimal objective value which is taken as the fitness of *l*. The distinct characteristic of the algorithm is that the bases of the follower's complementarity systems are searched instead of the variable values, which makes the search space finite.

This paper is organized as follows. The convex quadratic bilevel programming problem with transformation is presented in Section 2, and an evolutionary algorithm is given based on a local search technique in Section 3. Experimental results and comparison are presented in Section 4. We finally conclude our paper in Section 5.

2 Preliminaries

Consider the following nonlinear BLPP

$$\begin{cases} \min\limits_{x\geq 0} \; c_1^T x + d_1^T y + (x^T, y^T)Q_1(x^T, y^T)^T \\ s.t. \quad A_1 x + B_1 y \leq b_1, \text{ where y solves} \\ \min\limits_{y\geq 0} \; c_2^T x + d_2^T y + (x^T, y^T)Q_2(x^T, y^T)^T \\ s.t. \quad A_2 x + B_2 y \leq b_2 \end{cases} \tag{2.1}$$

Where Q_i are symmetric positive semidefinite matrices, $c_i \in R^n, d_i \in R^m \,(i = 1, 2)$; $b_1 \in R^p, b_2 \in R^q$; A_1, B_1, A_2, and B_2 are matrices with corresponding orders. For convenience, we assume the follower has only one solution for each selected *x* and the rank of B_2 is q.

In problem (2.1), let

$$Q_i = \begin{pmatrix} Q_x^i & Q_{xy}^i \\ Q_{xy}^{i\;T} & Q_y^i \end{pmatrix}$$

Where $Q_x^i \in R^{n\times n}$, $Q_y^i \in R^{m\times m}$ and $Q_{xy}^i \in R^{n\times m}\,(i = 1, 2)$. Then the follower's objective can be transformed into

$$f(x, y) = c_2^T x + x^T Q_x^2 x + (d_2 + 2Q_{xy}^{2\;T} x)^T y + y^T Q_y^2 y \tag{2.2}$$

Since for any fixed *x*, $c_2^T x + x^T Q_x^2 x$ is constant when $f(x, y)$ is optimized, and so we ignore these two terms in the follower's objective function. Further, the follower's problem can be re-written as

$$\begin{cases} \min\limits_{y\geq 0} y^T Q_y^2 y + c(x)y \\ s.t. \; B_2 y \leq b(x) \end{cases} \tag{2.3}$$

Where $c(x)^T = d_2 + 2Q_{xy}^{2\;T} x$ and $b(x) = b_2 - A_2 x$. Obviously, each component of both $c(x)$ and $b(x)$ is a linear function of *x*. Since for *x* fixed, the follower's problem is a convex quadratic programming with respect to *y*, K-K-T stationary point problem of the follower's programming can be written as

$$\begin{cases} v - 2Q_y^2 y - B_2^T u = c(x)^T \\ y_0 + B_2 y = b(x) \\ v^T y = 0, y_0^T u = 0 \\ u, v, y, y_0 \geq 0 \end{cases} \tag{2.4}$$

Where u and v are Langrangian multipliers and $y_0 \in R^q$ is a slack vector. (2.4) is also called the linear complementarity systems of the follower's programming. Further, let

$$w = \begin{pmatrix} v \\ y_0 \end{pmatrix}, \quad z = \begin{pmatrix} y \\ u \end{pmatrix}, \quad M = \begin{pmatrix} 2Q_y^2 & B_2^T \\ -B_2 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} c(x)^T \\ b(x) \end{pmatrix}$$

Then (2.4) can be rewritten as

$$\begin{cases} w - Mz = q & (c1) \\ w, z \geq 0 & (c2) \\ w^T z = 0 & (c3) \end{cases} \tag{2.5}$$

For problem (2.5), if (w, z) satisfying (c3) is a basic feasible solution of (c1-c2), then it is called the complementary basic feasible solution (CBFS) of (2.5)[3]. For fixed x, if one wants to obtain the optimal solution y of the follower's problem, all that needs to do is to solve problem (2.5) for a CBFS. In fact, it is computation-expensive to solve (2.5) directly. Considering the characteristics of solutions to (2.5), there exists at least one optimal base B for each selected x, which consists of some columns of the coefficient matrix (I, -M), and for each variable pair (w_i, z_i), $i = 1, 2, ..., m + q$, only one variable can be selected as basic variable according to (c3).

3 Solution Method

In this section, we present an EA for solving (2.1). At first, we encode each individual by considering the bases of (c1-c2) on condition that (c3) is satisfied. Then a new fitness function is given, which provides a local search for the variable values of the leader. Next we describe each step of the algorithm in more detail.

3.1 Chromosome Encoding

In this section the bases B of (2.5) are encoded as individuals, and each individual can be represented as a binary string $\bar{l}$ = 01001001...0 (the string length is 2(m+q)), where the total number of 1’s is m+q, and each 1 in $\bar{l}$ corresponds to a column of (I,-M) which is selected into B. But in the binary string the selection of some genes can be determined completely by the other genes when the constraint (c3) is considered, hence the genes of $\bar{l}$ can be reduced further. The matrix B can be taken as follows: At first, k ($0 \leq k \leq m+q$) columns are chosen randomly from I, and let the k column indices be $i_1, i_2, ..., i_k$. Then the remaining m+q-k columns can be taken from all columns of -M except for the $i_1 - i_k$ th columns. Noted that due to (c3), only m+q-k columns remain in –M which can be selected into B. It follows that the total m+q indices are completely determined by $i_1, i_2, ..., i_k$. Therefore, it is enough to represent B only using the columns selected from I.

According to the scheme, a chromosome can be denoted by a binary string $l = 01001001...0$ (m+q gene bits), where 1's represent the columns in I which belong to B, whereas 0's stands for the columns not in B. If B corresponding to l is singular, the chromosome is called an infeasible individual; otherwise, it is feasible.

3.2 Fitness Evaluation

For each individual l, if it is a feasible individual, the fitness $R(l)$ of l is given as following steps: At first, according to l, we solve (2.5) and get a CBFS, in which the follower's solution $y(x)$, as a linear function of x, can be obtained. Then we replace y in (2.1) by the $y(x)$ and take $y(x) \geq 0$, as a result, the leader problem can be converted to an optimization problem only involving x. At last, the problem is solved to obtain the optimal value F and the value is taken as the fitness of l. If the feasible region of the problem is empty, we take $R(l)$=M, where M is large enough. The solving process, in fact, is a local search, which can find the "best" one in all values of x for which $y(x)$ is the optimal solution to the follower's problem. If the individual l is infeasible, we also let $R(l) = M$. After doing so, it is convenient for us to encode all individuals.

3.3 Crossover and Mutation Operators

The crossover operator is given as follows. Let l' and l'' are parents for crossover and an offspring is denoted by o' . At first, we label all gene bits with the same values, then copy these values to the same gene bits of o' and let the amount of 1's in o' be d. We select randomly m+q-d genes in the remaining bits and set the related values be 1, whereas other values are taken as 0.

The bit mutation operator is adopted in the proposed algorithm.

3.4 Evolutionary Algorithm with Local Search (EA/LS)

In this subsection, we present an evolutionary algorithm with local search based on the encoding scheme, fitness function and evolutionary operators described above.

Step 1 (*Initialization*) Randomly generate N initial points, which form the initial population $pop(0)$ with population size N. Let k =0.

Step 2 (*Fitness*) Evaluate the fitness value of each point in $pop(k)$;

Step 3 (*Crossover*) Execute the crossover and get offspring set O_c ;

Sept 4 (*Mutation*) Execute the mutation and get its offspring set O_m ;

Table 4.1: Comparison of the results found by EA/LS and by the compared algorithms**.**

No.	$EA/LS - F(x,y)$					*Ref*− $F(x^*, y^*)$
	$F(x^*, y^*)$	F_{mean}	F_{median}	$F(\bar{x}, \bar{y})$	*STD*	
T01[5]	17	17	17	17	0	17
T02[10]	-11.999	-11.999	-11.999	-11.999	0	-11.999
T03[7]	231.25	231.25	231.25	231.25	2.4e-9	231.25
T04[7]	0.6389	0.6389	0.6389	0.6389	1.1e-8	0.6389
T05[10]	225	225	225	225	0	225
T06[1,10]	0	0	0	0	0	0[10], 5[1]
T07[10]	-7.58	-7.58	-7.58	-7.58	0	-7.58
T08[10]	-8.92	-8.92	-8.92	-8.92	0	-8.92
T09[9,10]	-3.92	-3.92	-3.92	-3.92	0	-3.92[10], -3.15[9]
T10[10]	-3.6	-3.6	-3.6	-3.6	0	-3.6

Step 5 (*Selection*) Evaluate the fitness values of all points in $O_c \cup O_m$. Select the best N_1 points from the set $pop(k) \cup O_c \cup O_m$ and randomly select $N - N_1$ points from the remaining points of the set. These selected points form the next population $pop(k+1)$.

Step 6 If the termination condition is satisfied, then stop; otherwise, let $k = k+1$, go to Step 3.

4 Simulation Results

We select 10 test problems from the references，which are convex and quadric except for T06 (linear-quadric BLPP). We use the active-set method (the simplex method for T06) to solve the resulting problems. The parameters are chosen as follows: the population size *N*=30, the crossover probability $p_c = 0.8$, the mutation probability $p_m = 0.1$, $N_1 = 20, \quad M = 10000$. The algorithm stops when the maximum of 50 generations is achieved. We execute EA/LS in 20 independent runs on each problem on a PC(Intel Pentium IV-2.66GHz), and record the following data: Leader's objective value $F(x^*, y^*)$ at the best solutions; leader's objective value $F(\bar{x}, \bar{y})$ at the worst solutions; mean value F_{mean} of the leader's objective values in 20 runs as well as median F_{median} and standard deviation *STD*.

All results are shown in Table 4.1. Table 4.1 provides the comparison of the results found by EA/LS in 20 runs and by the compared algorithms for T01-T10. Ref stands for the related algorithms in the references.

It can be seen from Table 4.1 that for problems T06[1] and T09[9], all results found by EA/LS in 20 runs are better than the best results given by the compared algorithms. For other problems, the best results found by EA/LS are as good as those by the compared algorithms. In all 20 runs, EA/LS found the best results of all problems except for T03 and T04. For these two problems, we can find the standard deviations are very small, which means that EA/LS is stable and robust.

5 Conclusion

For convex quadric bilevel programming problems, the paper develops an evolutionary algorithm based on a local search method. The technique avoids solving the follower's programming directly, and so the amount of computation is reduced. Obviously, the proposed algorithm can be used to solve linear-quadric bilevel programming in which LPs are solved instead of quadric programming.

Acknowledgements

This research work was supported by the National Natural Science Foundation of China (No.61065009, No. 60873099) and the Innovation Research Foundation of Qinghai Normal University.

References

[1] E. Aiyoshi, K. Shimuzu, A solution method for the static const rained Stackelberg problem via Penalty Method. IEEE Trans. Automat. Control 229(1984), 1112-1114.

[2] J. F. Bard, *Practical Bilevel Optimization*, Kluwer Academic Publishers, The Netherlands, 1998.

[3] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley, Chichester, 1979.

[4] B. Colson, P. Marcotte and G. Savard, Bilevel programming: A survey. 4OR 3(2005), 87-107.

[5] B. Colson, P. Marcotte and G. Savard, A trust-region method for nonlinear bilevel programming: algorithm and computational experience, Comput. Optim. Appl. 30(2005), 211-227.

[6] H. I. Calvete, C. Gale and P. M. Mateo, A new approach for solving linear bilevel problems using genetic algorithms. European J. Oper. Res. 188(2008), 14-28.

[7] J. B. E. Etoa, Solving convex quadratic bilevel programming problems using an enumeration sequential quadratic programming, J. Global Optim. Vol. 47(2010) 615-637.

[8] Kuen-Ming Lan, Ue-Pyng Wen and Hsu-Shih Shih, A hybrid neural network approach to bilevel programming problems, Appl. Math. Lett. 20(2007), 880-884.

[9] J. V. Outrata, One the numerical solution of a class of Stackelberg problems, Zeitschrift Fur Operation Research 34(1990), 255-278.

[10] Yuping Wang, Yong-Chang Jiao and Hong Li, An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme, IEEE Transactions on Systems, Man and Cybernetics 35(2005), 221-232.

Hecheng Li received the M.S. degree in mathematics from Northwest Normal University, Lanzhou, China, in 2001, and the PhD degree in Applied Mathematics from Xidian University, Xi’an, China, in 2009. He is currently a professor in Qinghai Normal University. His research interests include evolutionary computation, optimization theory, algorithms and applications.

Yuping Wang received the B.Sc. degree in mathematics from Northwest University, Xi’an, China, in 1983, and the Ph.D. degree in computational mathematics from Xi’an Jiaotong University, Xi’an, China, in 1993. Currently, he is a professor with the School of Computer Science and Technology, Xidian University, Xi’an, China. His research interests include evolutionary computation, optimization theory, algorithms, and applications.