Genetic Algorithm for Solving Convex Quadratic Bilevel Programming Problem*

Wang Guang-Min, Wan Zhong-Ping
School of Mathematics and Statistics,
Wuhan University, Wuhan 430072
Wang Xian-Jia
Institute of Systems Engineering
Wuhan University, Wuhan 430072

Abstract: This paper presents a genetic algorithm method for solving convex quadratic bilevel programming problem. Bilevel programming problems arise when one optimization problem, the upper problem, is constrained by another optimization, the lower problem. In this paper, the bilevel convex quadratic problem is transformed into a single level problem by applying Kuhn-Tucker conditions, and then an efficient method based on genetic algorithm has been proposed for solving the transformed problem. By some rule, we simplify the transformed problem, so we can search the optimum solution in the feasible region, and reduce greatly the searching space. Numerical experiments on several literature problems show that the new algorithm is effective in practice.

Key words: quadratic bilevel programming problem, genetic algorithm, optimal solution.

AMS (MOS) subject classifications: 90C30

1 Introduction

The bilevel programming problem is an optimization problem that is constrained by another optimization problem. This mathematical programming model arises when two independent decision makers, ordered within a hierarchical structure, have conflicting objectives. The decision maker at the lower level has to optimize its own objective function under the given parameters from the upper level decision maker, who, in return, with complete information on the possible reactions of the lower, selects the parameters so as to optimize its own objective function.

The quadratic bilevel programming (QBP) problem is an optimization model formulated as follows:

*Supported by the National Natural Science Foundation of China (70371032, 60274048) and the Doctoral Foundation in Ministry of Education of China (20020486035) .

$$
\begin{aligned}
&\max_{x\geq 0} F(x,y) = c_1^T x + d_1^T y + (x^T, y^T) R (x^T, y^T)^T \\
&\text{where } y \text{ solves} \qquad (1)\\
&\max_{y\geq 0} f(x,y) = c_2^T x + d_2^T y + (x^T, y^T) Q (x^T, y^T)^T \\
&s.t. \;\; Ax + By \leq b
\end{aligned}
$$

where $F(x,y)$, $f(x,y)$ are the objective functions of the leader and the follower, respectively. $c_1, c_2 \in R^{n_1}$, $d_1, d_2 \in R^{n_2}$, $A \in R^{m\times n_1}, B \in R^{m\times n_2}, b \in R^m$. $R, Q \in R^{(n_1+n_2)\times(n_1+n_2)}$ are symmetric matrices, respectively, $x \in R^{n_1}, y \in R^{n_2}$ are the decision variables under the control of the upper level and lower level, respectively.

The conventional solution approach to the bilevel programming problem is to transform the original two level problems into a single level one by replacing the lower level optimization problem with its Kuhn-Tucker optimization conditions. Branch-and-bound method [1-6], descent algorithms [11, 12], and evolutionary method [8, 10, 13, 14] have been proposed for solving the bilevel programming problems based on this reformulation.

In this paper we consider the problem (1) where $F(x,y)$ and $f(x,y)$ are convex functions. Unlike the linear case, this problem does not necessarily attain its optimal solution at an extreme point of the constrain region. We transformed the problem 1 into a single level problem by Kuhn-Tucker optimization conditions. For solving the latter problem, an efficient method based on genetic algorithm has been proposed in which each feasible chromosome represents a feasible solution and thereby reducing the search space significantly. In Section 2, the development of the algorithm including some concepts and theories are presented. The genetic algorithm for solving the quadratic bilevel programming problem is proposed in Section 3. The computational studies on several literature problems are reported for the efficiency of the proposed method in Section 4. We conclude the paper in Section 5.

2 The Development of Genetic Algorithm for Solving the QBP Problem

In this section the development of the algorithm are discussed. In the problem (1), let

$$
Q = \begin{bmatrix} Q_2 & Q_1^T \\ Q_1 & Q_0 \end{bmatrix}
$$

where $Q_0 \in R^{n_2\times n_2}, Q_1 \in R^{n_2\times n_1}, Q_2 \in R^{n_1\times n_1}$. Then $f(x,y)$ is transformed into

$$
f(x,y) = c_2^T x + x^T Q_2 x + (d_2 + 2Q_1 x)^T y + y^T Q_0 y
$$

Let $S = \{(x,y)\big| Ax + By \leq b, x, y \geq 0\}$ denote the constraint region of the QBP problem. In order to ensure that the problem (1) is well posed we make assumption that S is nonempty and compact. Let

Q_0 be a negative-definite matrix, so for each fixed x, there exists a unique solution to the following programming problem

$$\begin{aligned} \max_{x,y\geq 0} f(x,y) &= c_2^T x + x^T Q_2 x + (d_2 + 2Q_1 x)^T y + y^T Q_0 y \\ & s.t. \quad Ax + By \leq b \end{aligned} \tag{2}$$

and the problem (1) has a global optimum [5]. So, we recall that a point (x, y) is said to be feasible to the problem (1) if $(x, y) \in S$ and y is an optimal solution of the problem (2). Kuhn-Tucker conditions for the second level problem are derived and then the problem (1) is transformed into a single level problem of the form:

$$\begin{aligned} \max F(x,y) &= c_1^T x + d_1^T y + (x^T, y^T) R (x^T, y^T)^T \\ s.t. \quad & Ax + By + w = b \\ & 2Q_1 x + 2Q_0 y - B^T u + v = -d_2 \\ & u^T w = 0 \\ & v^T y = 0 \\ & x, y, u, v, w \geq 0 \end{aligned} \tag{3}$$

where $u \in R^m, v \in R^{n_1}$ are Kuhn-Tucker multipliers associated with the lower problem. So we have the following theorem

Theorem 1 [2]*: $f(x, y)$ is continuous and convex and a constraint qualification hold for the problem (2) with fixed x at x^*. Then a necessary and sufficient condition that is (x^*, y^*) solves the problem (1) is that there exist a $u^* \in R^m, v^* \in R^{n_1} \geq 0$, such that (x^*, y^*, u^*, v^*) is the optimal solution of the problem (3).*

Hence, we recall that a point (x^*, y^*) is called an optimal solution of the problem (1) (solution for short), if (x^*, y^*) is an optimal solution of the problem (3).

The genetic algorithms (GA) are search and optimization procedures motivated by natural principles and selection [7]. Because of its simplicity, minimal problem restrictions, global perspective, and implicit parallelism, GA have been applied to a wide variety of problem domains including engineering, sciences and commerce. Now, a new method based on genetic algorithm is proposed to solve problem (3). It is assumed that the problem consists of m constraints and the decision maker of the lower level has n_2 decision variables under his control. Each chromosome is described by using a string consisting of $m + n_2$ binary components in the proposed algorithm. The first m components are associated with the vector u and the remaining n_2 components are associated with the vector v.

This chromosome, according to following rule [8], transforms the problem (3) into the problem (4) below. If the value of the i^{th} component of the chromosome corresponding to u_i, which is the

i^{th} component of u, is equal to zero, then the variable u_i is also equal to zero; In addition, its complementary variable w_i, which is the i^{th} component of w, is greater than or equal to zero, otherwise u_i is greater than or equal to zero and its complementary variable w_i is equal to zero. On the other hand, if the j^{th} component corresponding to the variable v_j, which is the j^{th} component of v , is zero, then the value of variable v_j is equal to zero and its complementary variable y_j is greater than or equal to zero, otherwise v_j is greater than or equal to zero and its complementary variable y_j is equal to zero. This rule is applied with each chromosome for simplification of problem (3). The simplified problem is as follows:

$$\begin{aligned} \max F(x,y^{'}) &= c_1^T x + {d_1^{'}}^T y^{'} + (x^T, {y^{'}}^T)R^{'}(x^T, {y^{'}}^T)^T \\ s.t. \quad & Ax + B^{'}y^{'} + w^{'} = b \\ & 2Q_1 x + 2{Q_0}^{'}y^{'} - {B^{"}}^T u^{'} + v^{'} = -d_2 \\ & x, y^{'}, u^{'}, v^{'}, w^{'} \geq 0 \end{aligned} \tag{4}$$

In the problem (4), $y^{'}, u^{'}, v^{'}$, and $w^{'}$ are those component of y, u, v, and w that are greater than or equal to zero. Also, ${d_1}^{'}$ is the component of d_1 associated with $y^{'}$. The columns of the matrices $B^{'}$ and $B^{"}$ are the columns and rows of B, which are associated with the variables $y^{'}$ and $u^{'}$, respectively. The columns of the matrix ${Q_0}^{'}$ are the columns of Q_0, which are associated with the variables $y^{'}$. The columns of the matrix $R^{'}$ are the columns of R, which are associated with the variables $y^{'}$.

solution of the problem (4), if existing is a feasible solution for quadratic bilevel programming problem and the optimal value of the objective function is the fitness value for this chromosome.

The following theorem helps us to find a way to search and evaluate the feasible solution of the QBP problem.

Theorem 2 *If the solution of the simplified problem (4) by each chromosome exists, then this solution is a feasible solution of the original problem (1).*

3 The description of the proposed GA:

This yields the following genetic algorithm steps which are described in turn:

Step 1: Generating the initial population

The initial population consists of a set of feasible chromosomes. For generating these chromosomes the following problem is solved:

$$\begin{aligned} \max_{x,y\geq 0} f^{'}(x,y) &= c_2^T x + x^T r x + (d_2 + 2Q_1 x)^T y + y^T Q_0 y \\ s.t. \quad & Ax + By \leq b \end{aligned}$$

where $r \in R^{n_1 \times n_1}$ is a random matrix. By changing the components of r, the optimal solution also changes, but the optimality conditions hold. These feasible solutions are converted into chromosomes and the values of the objective functions of the first level are used for fitness value of each chromosome. After generating sufficient such chromosomes, then go to step 2.

Step 2: Crossover

In this step, firstly, a random number $P_c \in [0, 1]$ is generated. This number is the percentage of the population on which the crossover is performed. Then, two chromosomes are selected randomly from the population as the parents. Children are generated using the following procedure: Random integer c is generated in the interval $[1, l-1]$, where l is the number of components of a chromosome($l = m + n_2$). The c^{th} first components of the children are the same components as the respective parents (i.e. The first child from the first parent and the second child from the second parent.). The remaining components are selected according to the following rules:

i. The $(c+i)^{th}$ component of the first child is replaced by the $(l-i+1)^{th}$ component of the second parent (for $i = 1, 2, \cdots, l-c$).

ii. The $(c+i)^{th}$ component of the second child is replaced by the $(l-i+1)^{th}$ component of the first parent (for $i = 1, 2, \cdots, l-c$).

For example, by applying the proposed operator for the following parents, and assuming $c = 5$, we obtained the following children.

Parents	Children
10110 1100	10110 0100
11010 0010	11010 0011

Note that the proposed operator generates chromosomes with more variety since this operator can generate different children from similar parents. It is evaluated with problem (4) for feasibility and fitness value. If the new chromosome generated is unfeasible then it is eliminated and the algorithm continues. The crossover operation continues for P_c percent of the population.

Step 3: Mutation

In this step, firstly, a random number $P_m \in [0, 1]$ is generated. This number is the percentage of population on which the mutation is performed. Then one chromosome is selected randomly from the population. An integer random number n is generated in the interval $[1, l]$, where l is the length of the chromosome($l = m + n_2$) . For generating new chromosome the n^{th} component is changed to 0, if it was initially 1 and to 1 if it was initially 0. The new chromosome is evaluated with problem (4) for feasibility and fitness value. The mutation operation is performed for P_m percent of the population.

Step 4: Selection

The chromosomes are arranged in descending order of fitness value. A population corresponding to the size of the original population is selected from top of the list. This is considered as the new population.

Step 5: Termination

The algorithm terminates at the maximal iteration number. The best generated solution, which has been recorded in all iterations in the earliest time, is reported as the solution for QBP problem by proposed GA algorithm.

4 Computational Experiences

In order to test the efficiency of the proposed GA method, several literature problems are solved.
Example 1

$$\begin{aligned}
&\min (x_1-30)^2+(x_2-20)^2-20y_1+20y_2\\
&s.t. x_1+2x_2 \le 30\\
&\quad x_1+x_2 \ge 20\\
&\quad 0 \le x_1 \le 15\\
&\quad 0 \le x_2 \le 15\\
&\quad \min (x_1-y_1)^2+(x_2-y_2)^2\\
&\quad s.t.\\
&\qquad 0 \le y_1 \le 15\\
&\qquad 0 \le y_2 \le 15
\end{aligned}$$

Example 2[9]

$$\begin{aligned}
&\min {y_1}^2+{y_2}^2+x^2-4x\\
&s.t.\\
&\quad 0 \le x \le 2\\
&\text{where } y^T=(y_1,y_2) \text{ solves}\\
&\quad \min {y_1}^2+0.5{y_2}^2+y_1y_2+(1-3x)y_1+(1+x)y_2\\
&\quad s.t.\\
&\qquad 2y_1+y_2-2x \le 1\\
&\qquad y^T=(y_1,y_2) \ge 0
\end{aligned}$$

Example 3[9]

$$\begin{aligned}
&\min {y_1}^2+{y_3}^2-y_1y_3-4y_2-7x_1+4x_2\\
&s.t.\ x_1+x_2 \le 1\\
&\quad x=(x_1,x_2)^T \ge 0\\
&\text{where } y^T=(y_1,y_2) \text{ solves}\\
&\quad \min {y_1}^2+0.5{y_2}^2+0.5{y_3}^2+y_1y_2+(1-3x_1)y_1+(1+x_2)y_2\\
&\quad s.t.\\
&\qquad 2y_1+y_2-y_3+x_1-2x_2+2 \le 0\\
&\qquad y^T=(y_1,y_2,y_3) \ge 0
\end{aligned}$$

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

Prob. No.	result in the paper			result in the references		
	(x,y)	F	f	(x,y)	F	f
1	$(15,7.5,10,7.5)$	331.25	25	$(15,7.501,10,7.501)$	331.262	25
2	$(0.8462,0.7692,0)$	−2.0771	−0.5918	$(0.8438,0.7657,0)$	−2.0769	−0.5863
3	$(0.611,0.389,0,0,1.833)$	0.6389	1.6806	$(0.609,0.391,0,0,1.828)$	0.6426	1.6708

Notes: F and f are the objective function value of the upper-level and lower-level programming problem, respectively.

5 Conclusion

This paper presents a method based on genetic algorithm approach for solving bilevel programming problem. Kuhn-Tucker condition for the second level problem are derived and then the QBP problem is transferred into a single level problem with complementary constraints. By some rule in Ref [8], we simplify the transformed problem, so we can search the optimum solution in feasible region, and reduce greatly the searching space. From the numerical result, the results by the method in this paper accord with the results in the references. So the proposed GA is very efficient from computational point of view and quality of solutions

References

[1] F. Al-Khayyal, R. Horst, P. Paradalos. *Global Optimization of Concave Functions Subject to Quadratic Constraints: An Application in Nonlinear Bilevel Programming. Annals of Operations Research,* 1992. **34**: 125-147.

[2] J. F. Bard. *Convex Two-level Optimization. Mathematical Programming,* 1988. **40**: 15-27.

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

[4] J. F. Bard, J. T. Moore. *A Branch-and-Bound Algorithm for the Bilevel Programming Problem. SIAM Journal on Scientific and Statistical Computing,* 1990. **11(2)**: 281-292 .

[5] T. Edmunds, J. F. Bard. *Algorithms for Nonlinear bilevel mathematical programming. IEEE Transactions on Systems, Man, and Cybernetics,* 1991. **21**: 83-89.

[6] T. Edmunds, J. F. Bard. *An Algorithm for the Mixed-integer Nonlinear Bilevel Programming Problem. Annals of Operations Research,* 1992. **34**: 149-162.

[7] D.E. Golberg. *Genetic Algorithms in Search, Optimization and machine learning. Addison-Westey, Readomg, Mass,* 1989.

[8] S. R. Hejazi, A. Memariani, G. Jahanshanloo, M. M. Sepehri. *Bilevel Programming Solution by Genetic Algorithms. Proceedings of the First National Industrial Engineering Conference,* 2001.

[9] L. D. Muu, N. V. Quy. *A Global Optimization Method for Solving Convex Quadratic Bilevel Programming Problems. Journal of Global Optimization,* 2003. **26(2)**: 199–219.

[10] K. H. Sahin, A. R. Ciric. *A Dual Temperature Simulated Annealing Approach for Solving Bilevel Programming Problem. Computers and Chemical Engineering,* 1998. **23(1)**: 11–25.

[11] G. Savard, J. Gauvin. *The Steepest Descent Direction for the Nonlinear Bilevel Programming Problem. Operations Research Letters,* 1994. **15**: 265-272.

[12] L. Vicente, G. Savarg, J. Judice. *Descent Approaches for Quadratic Bilevel Programming. Journal of Optimization Theory and Applications,* 1994. **81(2)**: 379-399.

[13] Cang-Pu Wu. *Hybrid Technique for Global Optimization of Hierarchical Systems. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* 1996. **3**: 14–17.

[14] Yafeng Yin. *Genetic Algorithms Based Approach for Bilevel Programming Models. Journal of Transportation Engineering,* 2000. **126(2)**: 115-120.