Journal of Computations & Modelling, vol.3, no.4, 2013, 331-347
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2013

A Hybrid Approach Based on Genetic Algorithm for Mixed-Integer Bilevel Programming Problems

Hecheng Li[1]

Abstract

Bilevel programming problem is characterized as hierarchical structure, involving two optimization problems at deferent levels. When some variables are restricted into integer set, the problem is very challenging for most canonical optimization approaches. In the present paper, a class of nonlinear mixed-integer bilevel programs is taken into account in which the follower is an integer linear program, and a hybrid approach based on genetic algorithm is developed for solving the problems of this kind. Firstly, a genetic algorithm is used to explore the space of leader's variable values. Secondly, in order to obtain the optimal solution to the follower's problem, all potential bases of the follower's relaxed problem are determined and then the solution functions of the problem are presented by using these bases for distinct leader's variable values. Finally, if the solution to the follower's problem provided by the solution functions is not satisfy integer requirements, the follower is further solved by using traditional optimization

[1] Department of Mathematics, Qinghai Normal University, Xining 810008, China.
E-mail: hclwxy@aliyun.com

Article Info: *Received* : October 29, 2013. *Revised* : November 29, 2013.
Published online : December 1, 2013.

technique. Some computational examples are solved and the results show that the proposed algorithm is efficient and robust.

Mathematics Subject Classification: 90C11, 90C26, 90C30

Keywords: Bilevel programming problems, Genetic algorithm, Bases, Optimal solutions

1 Introduction

The bilevel programming problem (BLPP), as a mathematical model of the leader-follower game, been investigated extensively [1-3]. A bilevel programming problem consists of two optimization problems located at deferent levels with hierarchical structure, leader's problem and follower's problem. In such a problem, the feasible region of the leader's problem is implicitly determined by the follower's problem, and the decision variables are partitioned between the leader and the follower, each of whom optimizes his objective function. The general BLPP is of the form:

$$\begin{cases} \min\limits_{x\in X} F(x,y) \\ s.t. \quad G(x,y) \le 0 \\ \min\limits_{y\in Y} f(x,y) \\ s.t. \quad g(x,y) \le 0 \end{cases} \tag{1}$$

Where $x \in R^n$ and $y \in R^m$ are called the leader's and follower's variables, respectively, *F*(*f*) : $R^n \times R^m \to R$ is called the leader's (follower's) objective function, and the vector-valued functions G: $R^n \times R^m \to R^p$ and g: $R^n \times R^m \to R^q$ are called the leader's and follower's constraints, respectively; the sets *X* and *Y* place additional constraints on the variables, such as upper and lower bounds or integrality requirements [2]. Unlike other mathematical programs, the constraints of the bilevel programming problem always involve the optimality to the follower's problem. In BLPP(1), the leader first chooses a vector $x \in X \subseteq R^n$ in an attempt to optimize his/her objective function *F*(*x*, *y*). For this chosen *x*, the follower reacts by selecting a vector $y \in Y \subseteq R^m$ that optimizes his/her objective function *f*(*x*, *y*). If the point (x, y) satisfies all leader's constraints, the point is called a feasible point of BLPP(1). The optimal solution to BLPP(1) is best one among all feasible solutions, which makes the leader's objective minimized.

When some or all variables in (1) are restricted into integer set, the problem is known as a mixed-integer bilevel programming problem (MIBLPP). This kind of problems always arises when decision makers face a 'yes or no' decision problem, or the values of variables represent numbers of machines, products or people. From a computational viewpoint, the optimization problems with integer requirements are harder to solve than continuous ones, especially when the feasible region of problem is large. For MIBLPP, it seems to pose more algorithmic challenges than one-level mixed-integer optimization problems [5]. First, for a mixed-integer BLPP, even if the solution of the relaxed problem is an integer vector, it may not be a globally optimal solution of the original mixed-integer BLPP. In addition, the solution to the relaxed problem does not provide a valid lower bound for the solution of the original problem. As a result, there are only a small number of attempts to solve (mixed-) integer bilevel programming problems, and most of research on the mixed-integer BLPP is concentrated on a very restricted class of problems, such as linear BLPPs.

For integer linear bilevel programming problems, Moore and Bard developed a branch and bound type of implicit enumerative solution algorithm [5]. Vicente, Savard and Judice analyzed three classes of discrete linear bilevel programming problems, and presented some existence results of optimal solutions by which a penalty function method was developed for solving discrete linear bilevel programming problems [6]. Wen and Huang presented a simple tabu search algorithm to solve the mixed-integer linear bilevel programming problem only when the leader's variables involves integer requirement, and proposed two supplementary procedures [7]. Based on three weak assumptions, Xu and Wang proposed an exact algorithm for the mixed integer linear bilevel program and considered finite optimal, infeasible, and unbounded cases [8]. For nonlinear cases, parametric analysis technique is frequently adopted [9-12]. Dominguez and Pistikopoulos developed two algorithms for solving pure integer and mixed-integer bilevel programming problems by multi-parametric programming techniques [10]. However, additional parameters need to be added when problems to solve are not 0-1 case. Gümüs and Floudas presented a global optimization method to solve mixed-integer nonlinear BLPPs, in which the mixed-integer problem located at follower's level is transformed into a linear programming [11]. Jan and Chern proposed an algorithm using parametric analysis to deal with a special kind of nonlinear integer BLPPs in which functions need to be separable and monotone [12]. In addition, various intelligent methods have been applied to solve (mixed-) integer bilevel programs [13-15], especially when nonlinear or non-convex functions are involved [14].

Some real-world problems modeled as mixed-integer bilevel programs have also been investigated, and for these problems some efficient approaches have been developed [13, 16-17].

In this work, a class of mixed-integer nonlinear bilevel programming problems is discussed, in which only the follower's problem is a linear mixed-integer program. We apply a real-coded genetic algorithm to search the space of the leader's variable values. In order to obtain the follower's variable value for any x fixed, the relaxed problem of the follower, a linear program, is taken into account. First, all potential bases are taken and all solutions to the relaxed problem are represented as functions of leader's variables by using these bases. In addition, if the solution to the relaxed problem don't satisfy integer requirement for some selected x, the branch-and-bound approach is utilized to further solve the follower's problem.

This paper is organized as follows. Some notations and discussed problem are stated in Section 2, and Section 3 gives the follower's solution. In Section 4 we present genetic operators as well as algorithmic approach. Section 5 analyzes the convergence of the algorithm, and in Section 6 some computational examples are given and solved. We finally conclude our paper in Section 7.

2 Preliminaries

The mixed-integer nonlinear BLPP can be represented as

$$\begin{cases} \min_x F(x_I, x_C, y_{I'}, y_{C'}) \\ s.t.\ G(x_I, x_C, y_{I'}, y_{C'}) \le 0; \\ \min_y f(x_I, x_C, y_{I'}, y_{C'}) \\ s.t.\ g(x_I, x_C, y_{I'}, y_{C'}) \le 0; \\ x_I, y_{I'} \ \text{integer} \end{cases} \tag{2}$$

Where $x = (x_I, x_C) \in R^n$, $y = (y_{I'}, y_{C'}) \in R^m$; I and I' stand for the index sets of the integer variables and C and C' are those of the continuous variables at both levels; $F, f: R^n \times R^m \to R$, $G(g): R^n \times R^m \to R^p(R^q)$; Just like one-level mixed-integer programs, when the integrality constraints in (2) are deleted, the resulting problem is called the relaxed problem associated with the original one.

Basic Notations are listed as follows

a) Constrained region: $S = \{(x,y) \mid G(x,y) \le 0, g(x,y) \le 0, x_I, y_{I'} \ \text{integer}\}$;

b) For x fixed, follower's feasible set: $S(x) = \{y \mid g(x,y) \le 0, y_{I'} \ \text{integer}\}$;

c) Projection of S onto the leader's decision space:

$$S(X) = \{x \mid \exists y, \text{such that } (x,y) \in S\};$$

d) For each $x \in S(X)$, follower's rational reaction set:

$$M(x) = \{y \mid y \in \arg\min\{f(x,v), v \in S(x)\}\};$$

e) Inducible region: $IR = \{(x, y) \in S \mid y \in M(x)\}$.

In this work, we discuss MIBLPP (2) when the follower is a linear mixed-integer program, i.e., the discussed problem can be reformulated as follows：

$$\begin{cases} \min\limits_{x} F(x_I, x_C, y_{I'}, y_{C'}) \\ s.t.\ G(x_I, x_C, y_{I'}, y_{C'}) \le 0; \\ \min\limits_{y \ge 0} d_1 y_{I'} + d_2 y_{C'} \\ s.t.\ A_1 x_I + A_2 x_C + B_1 y_{I'} + B_2 y_{C'} \le b; \\ x_I, y_{I'} \text{ integer} \end{cases} \tag{3}$$

Here, A_i, B_i, d_i and $b \in R^q$ are of conformal dimensions, $i = 1,2$.

For leader's variable x fixed, the problem

$$\begin{cases} \min\limits_{y \ge 0} d_1 y_{I'} + d_2 y_{C'} \\ s.t.\ A_1 x_I + A_2 x_C + B_1 y_{I'} + B_2 y_{C'} \le b; \\ y_{I'} \text{ integer} \end{cases} \tag{4}$$

is called the follower's problem of (3). Furthermore, the problem

$$\begin{cases} \min\limits_{y \ge 0} dy \\ s.t.\ Ax + By \le b; \end{cases} \tag{5}$$

is known as the relaxed problem of the follower's problem, where $d = (d_1, d_2)$, $A = (A_1, A_2)$ and $B = (B_1, B_2)$. Since any inequality constraint can be transformed into equality by adding a slack variable, without any loss of generality, hence, (3) can be reformulated as

$$\begin{cases} \min\limits_{x} F(x_I, x_C, y_{I'}, y_{C'}) \\ s.t.\ G(x_I, x_C, y_{I'}, y_{C'}) \le 0; \\ \min\limits_{y \ge 0} d_1 y_{I'} + d_2 y_{C'} \\ s.t.\ A_1 x_I + A_2 x_C + B_1 y_{I'} + B_2 y_{C'} = b; \\ x_I, y_{I'} \text{ integer} \end{cases} \tag{6}$$

Also, the relaxed problem (5) can be rewritten as

$$\begin{cases} \min\limits_{y\geq 0} dy \\ s.t.\ By = b - Ax; \end{cases} \tag{7}$$

Which is a linear programming problem with parameter x. For the purpose of computational convenience, we further assume that the row rank of B is q.

3 Solutions to the Follower's Problem

For a fixed x, we attend to solve the follower's problem. Since the relaxed problem can provide a lower bound for one-level mix-integer program, as a result, (7) is first considered.

Let $B^1, B^2, \cdots, B^k$ are all potential optimal bases of (7). It means that B^i at least should satisfy:

1) Nonsingular;
2) There exists at least one $x \in S(X)$ such that $(B^i)^{-1}(b - Ax) \geq 0$, and
3) $d_{N^i} - d_{B^i}(B^i)^{-1}N^i \geq 0$, here, $d = (d_{N^i}, d_{B^i})$, $B = (B^i, N^i)$.

It follows that for each x, the solution to the relaxed problem of the follower can be represented as one of $y = (y_{B^i}, 0), i = 1,2,\cdots,k,$ here, $y_{B^i} = (B^i)^{-1}(b - Ax)$ is basic component. Furthermore, if this solution don't satisfy the integer requirement, the branch and bound method can applied to solve (4) with a starting solution y.

4 Proposed Algorithmic Approach

In the section, we begin with initial population, fitness function, crossover and mutation operators, and then propose a genetic algorithm using real number encoding to solve (3).

4.1 Initial Population

N points are generated randomly in *X*. If some entries don't satisfy integer restrictions, the entries are round into the nearest integers. As a result, a set, as an initial population denoted by pop(0), is obtained, satisfying all integer requirements on the leader's variables.

4.2 Fitness Evaluation

For each $x \in pop(0)$, the inequalities $(B^i)^{-1}(b - Ax) \geq 0$, $i = 1,2,\cdots,k$, are checked, one by one. If the s-th inequality is satisfied, the basic components of the optimal solution y can be represented as $(B^s)^{-1}(b - Ax)$ while other components are taken as 0. If this y don't satisfy the integer restriction, then the follower is further solved by using the branch and bound method with a lower bound *dy* of the objective function.

Once the optimal solution is obtained, a fitness function with a penalty term is presented as follows:

$$R(x, y) = F(x, y) + M \max\{0,\ G_i(x, y), i = 1,..., p\} \tag{8}$$

Here, M is a penalty parameter large enough.

4.3 Crossover and Mutation Operators

Let p_b denote the best one found so far and p_i be a selected individual for crossover. The crossover offspring of the individual is gotten as follows.

$$o_i = p_i + r(p_b - p_i) \tag{9}$$

Where r is randomly taken in intervals [0, 1] and [1, 2], respectively. For each *r*, one offspring is generated. Hence, 2 offspring can be obtained for each crossover individual.

Gauss mutation with the mean value of 0 is adopted in this work.

4.4 Hybrid Genetic Algorithm

In this subsection we present a hybrid genetic algorithm based on the solution function of the follower' problem (HGA/SF) as follows:

Step1. (*Initialization*) Randomly generate N initial points $x_i \in X$, $i = 1,2,..., N$. These points form the initial population pop(0) with population size N. Let k = 0;

Step2. (*Fitness*) Evaluate the fitness value R(x, y) of each point in pop(k);

Step3. (*Crossover*) Select parent individual *x* from pop(k) according to crossover probability p_c. For each selected parent *x*, execute the crossover. Let O1 stand for the set of all crossover offspring;

Step4. (*Mutation*) Select parents from O1 according to the mutation probability p_m. For each selected parent *x*, execute the mutation for *x* and get its offspring. Let O2 represent the set of all these offspring.

Step5. (*Selection*) Let $O = O1 \cup O2$. We evaluate the fitness values of all points in O, select the best N1 points from the set $pop(k) \cup O$ and randomly select $N - N_1$ points from the remaining points of the set. These selected points form the next population pop(k + 1);

Step6. If the termination condition is satisfied, then stop; otherwise, let k = k+1, go to Step3.

5 Convergent Analysis

In the proposed algorithm, it should be noted that once the optimal leader's variable values are determined, the corresponding follower's values can be obtained. As a result, we simply consider the leader's problem. In addition, we denote $R(x,y)$ by $\hat{R}(x)$.

In order to conveniently apply existing convergence results, some general assumptions are considered as follows [18]:

(A1) The search is executed in a bounded space $\hat{S}$;

(A2) $X^* = \{x^* \in \hat{S} \mid \hat{R}(x^*) = \min_{X \in \hat{S}} \hat{R}(x)\} \neq \Phi$;

(A3) For $\forall \varepsilon > 0$, let $M_\varepsilon = \{x \in \hat{S} \mid \hat{R}(x) - \hat{R}(x^*) \le \varepsilon, x^* \in X^*\}$, and the Lebesgue measure of the set is positive.

In addition, we denote the fitness values of the points in M_ε by R^* and the best fitness by R_t^* at the t-th generation. Let $D_t = R_t^* - R^*$.

Definition 5.1[18] *For* $\forall \delta > 0$, *if inequality* $\sum_{t=0}^{+\infty} P\{D_t \ge \delta\} < +\infty$ *holds, the evolutionary algorithm generating this sequence is convergent completely.*

According the convergence results in [18], if the proposed algorithmic approach satisfies: (a) the best individual in $pop(k) \cup O$ is directly put into the next generation of population pop(k + 1), and (b) the survival probability of each individual in $pop(k) \cup O$ has a common lower-bound p_s, then the algorithm is convergent completely. In fact, all points in $pop(k) \cup O$ are divided into two classes, i.e., the best N1 points and other points. All points in the first class including the best one are put into the next generation of population. For the second class, we set that the number of remaining points is v_t, then it is obvious that $v_t \le 3N$. It follows that the survival probability of each point in the second class is at least $(N - N_1)/3N$. It means the proposed algorithm satisfies these conditions. As a consequence, we have

Theorem 5.1 *The proposed HGA/SF is convergent completely.*

6 Computational experiment

In the simulation section, 5 test problems denoted by T1-T5, are selected, which are often adopted as benchmark problems to illustrate the efficiency of proposed approaches in literature. In our proposed algorithm, simply linearity is required in the follower's problem. It follows that the discussed model is more general than mixed-integer linear bilevel programs. These problems are as follows:

T1) [10-11]

$$\begin{cases} \min\limits_{x,y} \ (x-2)^2 + (y-2)^2 \\ \min\limits_{y} \ y^2 \\ s.t. \quad -2x - 2y \le -5, \\ x - y \le 1, \\ 3x + 2y \le 8, \\ x \in R, y \in [0,1], y \in Z \end{cases}$$

The problem is a mixed-integer nonlinear bilevel programming problem and the known best solution is (x, y)=(4/3,2) with F=4/9. In spite of the fact that the follower's problem is nonlinear, it should be noted that when the $f(x, y) = y^2$ is replace with $f(x, y) = y$, these two problems have the same optimal solutions. As a result, the original problem can be transformed into a MIBLPP in which the follower's problem is linear and solved by HGA/SF.

T2) [10-11]

$$\begin{cases} \min\limits_{x,y} x + 2y \\ \min\limits_{y} - y \\ s.t. - x + 2.5y \le 3.75, \\ - x - 2.5y \le -3.75, \\ 2.5x + y \le 8.75, \\ x, y \ge 0, x, y \in Z \end{cases}$$

The problem is an integer bilevel program and the best solution provided by the existing algorithm is (x, y)=(3,1) with F=5.

T3) [5, 10-11]

$$\begin{cases} \min\limits_{x,y} -x-10y \\ \min\limits_{y}\ y \\ s.t. -25x+20y \le 30, \\ x+2y \le 10, \\ 2x-y \le 15, \\ -2x-10y \le -15, \\ x, y \ge 0,\ x, y \in Z \end{cases}$$

The problem is also an integer bilevel program and the known best solution is (x, y)=(2,2) with F=-22.

T4) [8]

$$\begin{cases} \max\limits_{x,y}\ \ x-y \\ s.t.\ x \ge 1, \\ y \ge 2x, \\ \min\limits_{y}\ \ y \\ s.t.\quad y \ge x, \\ y \ge 0,\ y \in Z \end{cases}$$

The problem is infeasible and was given in reference [8] as a counterexample to illustrate the presented algorithm can't be terminated in a finite number of steps when the leader's variables are unbounded or have a large value region.

Table 1: Comparisons of the objective values

No.	F_{best}	F_{wost}	F_{mean}	std	F_{known}
1	0.4444	0.4446	0.4444	1.1e-5	0.4444
2	5	5	5	0	5
3	-22	-22	-22	0	-22
4	infeasible	infeasible	infeasible	-	infeasible
5	-1011.67	-1011.67	-1011.67	1.6e-3	-1011.67

Table 2: Computational cost and best solutions

No.	CPU(s)	MNI	Solutions
1	44.1	4.3e3	(1.333,2)
2	0.50	179	(3,1)
3	0.30	173	(2,2)
4	0.51	198	infeasible
5	0.14	189	(0,1,0,1,0,75,21.67)

T5) [9]

$$\begin{cases} \min\limits_{x,y} -(20x_1 + 60x_2 + 30x_3 + 50x_4 + 15y_1 + 10y_2 + 7y_3) \\ \min\limits_{y} \ -(20y_1 + 60y_2 + 8y_3) \\ s.t. \quad 5x_1 + 10x_2 + 30x_3 + 5x_4 + 8y_1 + 2y_2 + 3y_3 \le 230, \\ 20x_1 + 5x_2 + 10x_3 + 10x_4 + 4y_1 + 3y_2 \le 240, \\ 5x_1 + 5x_2 + 10x_3 + 5x_4 + 2y_1 + y_3 \le 90, \\ x \in \{0,1\},\ y \ge 0 \end{cases}$$

In this problem only the leader's variables are restricted into integer set. The optimal solution is $(x, y) = (0, 1, 0, 1, 0, 75, 21.67)$ with F=-1011.67.

The parameters are chosen as follows: the population size N = 50 for T1 and T4, whereas N=10 for other problems; the crossover probability pc = 0.8, the mutation probability pm = 0.1, N1 = [N/2], M = 10000. For T1–T5, the algorithm stops while the best results are not improved within 10 continuous generations or after 50 generations. We execute EDGA in 20 independent runs on each problem on a computer (Intel® Cor™2 Duo CPU E8400 2.99GHz), and record the following data:

(1) Best solutions (x^*, y^*) and leader's objective values F_{best} at the best points.

(2) Leader's objective function values F_{worst} at the worst solutions.

(3) Mean values (F_{mean}) and standard deviations (Std) of objective functions $F(x, y)$ in 20 runs.

(4) Mean values of CPU time (CPU) in 20 runs and mean number of individuals evaluated by HGA/SF (denoted by MNI).

All results are presented in Tables 1-2, in which Table 1 provides the comparison of the objective values found by HGA/SF in 20 runs and the compared algorithms. Table 2 shows CPU, MIN and the best solutions found by HGA/SF in 20 runs.

It can be seen from Table 1 that for all test problems, HGA/SF can find the best (optimal) results.

In all 20 runs, HGA/SF found uniformly the results of all problems since the standard deviations std of those problems are close to zero. This means that HGA/SF is stable and robust for these test problems.

In Table 2, from MIN and CPU, one can see easily that HGA/SF found the best solutions of these problems.

In spite of the fact that the proposed algorithm has obtained the same computational results as those presented in compared references, some advantages of HGA/SF should be noted when comparing the performance of these approaches: (i) the algorithms presented in references [5] and [8] can only deal with linear bilevel programming problems with integer variables. When the leader's problem involves nonlinear terms, these algorithms are applied directly, for example, problem T1. (ii) In references [9-11] parametric programming method is used to deal with integer variables and nonlinear terms, but the procedure cause additional variables generated, which increases the computational cost.

7 Conclusion

In presented paper a hybrid approach based on genetic algorithm has been developed for solving nonlinear mixed-integer bilevel programs with linear lower-level problems. In proposed approach genetic algorithm is used to explore the space of leader's variable values, and the solution functions of the follower are obtained in advance, which avoids frequently solving the lower level problem. The discussed problem extended linear cases to nonlinear ones and no additional variables are involved.

ACKNOWLEDGEMENTS. The research work was in part supported by the National Natural Science Foundation of China under Grant No. 61065009 and the Natural Science Foundation of Qinghai Provincial under Grant No. 2013-z-937Q.

References

[1] B. Colson, P. Marcotte and G. Savard, Bilevel programming: a survey, *A quarterly journal of operations research (4OR)*, 3(2), (2005), 87-107.

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

[3] Ue-Pyng Wen and Shuh-Tzy Hsu, Linear bi-level programming problems-a review, *Journal of the operational research society*, **42**(2), (1991), 125-133.

[4] Yuping Wang, Hong Li and Chuangyin Dang, A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence, *INFORMS Journal on computing*, **23**(4), (2011), 618-629.

[5] J.T. Moore and J.F. Bard, The mixed integer linear bilevel programming problem, *Operations research*, **38**(5), (1990), 911-921.

[6] L. Vicente, G. Savard and J. Judice, Discrete linear bilevel programming problem, *Journal of optimization theory and applications*, **89**(3), (1996), 597-614.

[7] U.P. Wen and A.D. Huang, A simple tabu search method to solve the mixed-integer linear bilevel programming problem, *European journal of operational research*, **88**(3), (1996), 563-571.

[8] Pan Xu and Lizhi Wang, An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions, *Computers & operations research*, **41**, (2014), 309–318.

[9] N.P. Faisca, V. Dua, B. Rustem, P.M. Saraiva and E.N. Pistikopoulos, Parametric global optimization for bilevel programming, *J. global optimization*, **38**(4), (2007), 609-623.

[10] L. F. Dominguez and E. N. Pistikopoulos, Multiparametric programming based algorithms for pure integer and mixed-integer bilevel programming problems, *Computers and chemical engineering*, **34**, (2010), 2097-2106.

[11] Z. H Gümüs and C. A. Floudas, Global optimization of mixed-integer bilevel programming problems, *Computational management Science*, **2**(3), (2005), 181–212.

[12] R.H. Jan and M.S. Chern, Nonlinear integer bilevel programming, *European journal of operational research*, **72**, (1994), 574-578.

[13] H.I. Calvete, C. Gale, and M.J. Oliveros, Bilevel model for production–distribution planning solved by using ant colony optimization, *Computers & operations research*, **38**, (2011), 320-327.

[14] H. Li and Y. Wang, Exponential distribution-based genetic algorithm for solving mixed-integer bilevel programming problems, *Journal of systems engineering and electronics*, **19**(6), (2008), 1157–64.

[15] I. Nishizaki, M. Sakawa, and T. Kan, Computational methods through genetic algorithms for obtaining Stackelberg solutions to two-level integer

programming problems, *Electronics and communications in Japan*, **86**(6), (2003), 59-66.

[16] Z. Lukac, K. Soric and V. Rosenzweig, Production planning problem with sequence dependent setups as a bilevel programming problem, *European journal of operational research*, **187**, (2008), 1504-1512.

[17] S. Dempe, V. Kalashnikov and R. Z. Rios-Mercado, Discrete bilevel programming: Application to a natural gas cash-out problem, *European journal of operational research*, **166**, (2005), 469-488.

[18] Yuping Wang, *Theory and methodology of evolutionary computation*, Science Press, Beijing, 2011. (In Chinese)