Available online at www.tjnsa.com
J. Nonlinear Sci. Appl. 9 (2016), 3672–3685
Research Article

An improved simulated annealing algorithm for bilevel multiobjective programming problems with application

Tao Zhang[a], Zhong Chen[a], Yue Zheng[b], Jiawei Chen[c,d,*]

[a]*School of Information and Mathematics, Yangtze University, Jingzhou 434023, China.*
[b]*School of Management, Huaibei Normal University, Huaibei 235000, China.*
[c]*School of Mathematics and Statistics, Southwest University, Chongqing 400715, China.*
[d]*College of Computer Science, Chongqing University, Chongqing 400044, China.*

Communicated by Y. H. Yao

Abstract

In this paper, an improved simulated annealing (SA) optimization algorithm is proposed for solving bilevel multiobjective programming problem (BLMPP). The improved SA algorithm uses a group of points in its operation instead of the classical point-by-point approach, and the rule for accepting a candidate solution that depends on a dominance based energy function is adopted in this algorithm. For BLMPP, the proposed method directly simulates the decision process of bilevel programming, which is different from most traditional algorithms designed for specific versions or based on specific assumptions. Finally, we present six different test problems to measure and evaluate the proposed algorithm, including low dimension and high dimension BLMPPs. The experimental results show that the proposed algorithm is a feasible and efficient method for solving BLMPPs.

Keywords: Bilevel multiobjective programming, simulated annealing algorithm, Pareto optimal solution, elite strategy.
2010 MSC: 65K10, 90C25.

1. Introduction and Preliminaries

Bilevel programming problem (BLPP) arises in a wide variety of scientific and engineering applications including optimal control, process optimization, game-playing strategy development, transportation problem

*Corresponding author
Email address: J.W.Chen713@163.com (Jiawei Chen)

Received 2016-02-15

and so on. In the past years, the theory and method of BLPP has been extensively studied, see [3–7, 14, 15, 22, 28, 33, 34, 36, 37] and the references therein. It is worth noting that the evolutionary algorithm (EA) is an important method for solving BLPP, see [18, 21, 25, 35, 41] and the references therein.

Since the multiobjective characteristics widely exist in many practical problems, it is necessary and interesting to study numerical method for solving bilevel multiobjective programming problems (BLMPP). An interactive algorithm was proposed for BLMPP, see [1, 24, 26, 27]. Eichfelder [16, 17] presented numerical methods for solving nonlinear bilevel multiobjective optimization problems with coupled upper level constraints and for solving nonlinear non-convex bilevel multiobjective optimization problems. Zhang and Lu [43] developed an approximation Kth-best algorithm to solve multi-follower (cooperative) bilevel programming problems with fuzzy multi-objective. In recent years, the metaheuristic has attracted considerable attentions as an alternative method for BLMPP. Recently, Deb and Sinha discussed BLMPP based on evolutionary multiobjective optimization principles, see [10–12, 30]. Based on those studies, Deb and Sinha [13] also introduced a viable and hybrid evolutionary-local-search based algorithm, and presented challenging test problems. Sinha [29] presented a progressively interactive evolutionary multiobjective optimization method for BLMPP. Very recently, Zhang et al. [42] proposed a hybrid particle swarm optimization algorithm with crossover operator to solve high dimensional BLMPP.

Simulated annealing (SA) [20] is a stochastic optimization method that is based on an analogy with physical annealing, which has been found to be quite successful in a wide variety of optimization tasks. Initially, SA has been used with combinatorial optimization problems [23]. Afterwards, SA has been extended to the single and multiobjective optimization problems with continuous N-dimensional control spaces [2, 31, 32]. Due to its robust and relative simplicity, there have been a few attempts in solving BLPP by SA. For example, Sahin and Ciric [19] proposed a dual temperature SA algorithm for solving BLPP. Xiang et al. [39] presented a method to solve bilevel optimization model on high-speed railway station location using SA algorithm. However, it is worth noting that the mentioned paper above only for bilevel single objective problems and the BLMPP has seldom been studied using SA so far.

In this paper, an improved SA is proposed for solving BLMPP. For such problems, the proposed algorithm directly simulates the decision process of bilevel programming, which is different from most traditional algorithms designed for specific versions or based on specific assumptions. The BLMPP is transformed to solve multiobjective optimization problems in the upper level and the lower level interactively by the improved SA. For one time of iteration, a set of approximate Pareto optimal solutions for BLMPP is obtained using the elite strategy. This interactive procedure is repeated until the accurate Pareto optimal solutions of the original problem are found. The rest of the paper is organized as follows. In Sect. 2, the BLMPP is provided. The proposed algorithm for solving bilevel multiobjective problem is presented in Sect. 3. In Sect. 4, some numerical examples are given to demonstrate the feasibility and efficiency of the proposed algorithm. The proposed algorithm is applied for solving a practical problem in Sect. 5, while the conclusion is reached in Sect. 6.

2. Problem Formulation

Let $X \in R^{n_1}, Y \in R^{n_1}$, $F : R^{n_1} \times R^{n_2} \to R \in R^{m_1}$, $f : R^{n_1} \times R^{n_2} \to Y \in R^{m_2}$, $G : R^{n_1} \times R^{n_2} \to R \in R^p$, $g : R^{n_1} \times R^{n_2} \to R \in R^q$. The general model of the BLMPP can be written as follows:

$$\begin{aligned} \min_{x,y} \quad & F(x,y) \\ s.t. \quad & G(x,y) \geqslant 0 \\ & \min_{y} \quad f(x,y) \\ & s.t. \quad g(x,y) \geqslant 0, \end{aligned} \tag{2.1}$$

where $F(x,y)$ and $f(x,y)$ are the upper level and the lower level objective functions, respectively. $G(x,y)$ and $g(x,y)$ are the upper level and the lower level constraints, respectively.

Let $S = \{(x,y)|G(x,y) \geqslant 0 g(x,y) \geqslant 0\}$, $X = \{x|\exists y, G(x,y) \geqslant 0, g(x,y) \geqslant 0\}$, and for the fixed $x \in X$, we denote by $\overline{S}(X)$ the weak efficiency set of solutions to the lower level problem, the feasible solution set of problem (2.1) is denoted by $IR = \left\{S = (x,y)|(x,y) \in S, y \in \overline{S}(X)\right\}$.

Definition 2.1. For a fixed $x \in X$, if y is a Pareto optimal solution to the lower level problem, then (x, y) is a feasible solution to the problem (2.1).

Definition 2.2. If (x^*, y^*) is a feasible solution to the problem (2.1), and there is no $(x, y) \in IR$, such that $F(x, y) \prec F(x^*, y^*)$, then x^*, y^* is a Pareto optimal solution to the problem (2.1), where " $\prec$ " denotes Pareto preference.

For problem (2.1), it is noted that a solution (x^*, y^*) is feasible for the upper level problem if and only if y^* is an optimal solution for the lower level problem with $x = x^*$. In practice, we often make the approximate Pareto optimal solutions of the lower level problem as the optimal response feed back to the upper level problem, and this point of view is accepted usually. Based on this fact, the SA algorithm may have a great potential for solving BLMPP. On the other hand, unlike the traditional point-by-point approach mentioned in Sect. 1, the improved SA algorithm apply a group of points in its operation, thus the improved SA can be developed as a new way for solving BLMPP. In the following, we present an improved SA algorithm for solving problem (2.1).

3. The Algorithm

3.1. The improved SA algorithm

The SA algorithm realizes the combination of the local search and global search through the cooling schedule. The cooling schedule is critical to the performance of the algorithm. The candidate solution is usually created by the current solution with a random perturbed vector, the probability density function of the random perturbed vector and the accept probability of the corresponding candidate solution are in relation to the temperature. When the temperature is higher, the search range of the candidate solution is wider, and it can also be accepted easily. When temperature is lower, the candidate solutions are constrained in the local area of the current solution, the search become local exploration. In this study, in order to improve the global search ability of the SA algorithm, the random perturbation vector is constructed based on cooling schedule used in [40] and the global convergence can be guaranteed. On the other hand, the method for computing the energy difference between the current solution and the candidate solution of multiobjective optimization problem used by [31] is employed in this paper.

Suppose the current solution is X^k, the candidate solution is Y^k, for the $k-th$ iteration at temperature T_k, the particle is updated by the improved SA as following:

Algorithm 1:

Step1 Create candidate solution according to the current solution.

$$Y^k = X^k + \varepsilon.$$

Step2 Compute energy difference between the current solution and the candidate solution.

$$\Delta E(X^k, Y^k) = \frac{1}{|\overline{F}|}(|F_{X^k}| - |F_{Y^k}|), \eta = random\,(0, 1)\,.$$

Step3 Compute transition probability.

$$p(Y^k|X^k, T_k) = min\left\{1, exp(\Delta E(X^k, Y^k))\right\}.$$

Step4 Choose the offspring. If $p(Y^k|X^k, T_k) \geqslant \eta$, $X^{k+1} = Y^k$; Otherwise $X^{k+1} = X^k$.

In Step1, $\varepsilon^k = (\varepsilon_1^k, \varepsilon_2^k, \cdots, \varepsilon_n^k)$, and the component ε_i^k of the random perturbed vector ε^k is produced by $\varepsilon_i^k = sign(U_i)T_k\left(\frac{1}{|U_i|^m} - 1\right)$, $(i = 1, 2, \cdots, n)$ where $U_1, U_2, \cdots, U_n \in random\,(-1, 1)$ are mutual independence uniform distribution random variables, $sign(\cdot)$ is the sign function, $m\,(m \geqslant 1)$ is the predefined

constant. In Step 2, F is the approximate Pareto front, which is the set of mutually non-dominating solutions found thus far in the annealing. Denote by $\overline{F}$ the union of the F, the current solution X^k and the proposed solution Y^k, that is, $\overline{F} = F \bigcup X^k \bigcup Y^k$. We denote by F_{X^k} the elements of F that dominate X^k and F_{Y^k} the elements of F that dominate Y^k.

3.2. The algorithm for solving BLMPP

The process of the proposed algorithm for solving BLMPP is an interactive co-evolutionary process. We first initialize population, and then solve multiobjective optimization problems in the upper level and the lower level interactively using the improved SA algorithm. For one time of iteration, a set of approximate Pareto optimal solutions for problem (2.1) is obtained by the elite strategy adopted in Deb et al. [9]. This interactive procedure is repeated until the accurate Pareto optimal solutions of problem (2.1) are found. The details of the proposed algorithm are given as following:

Algorithm 2:

Step 1. Initializing.

Step 1.1. Initialize the population P_0 with N_u particles which is composed by $n_s = N_u/N_l$ sub-swarms of size N_l each. The particles position is presented as: $z_j = (x_j, y_j)\,(j = 1, 2, \cdots, n_l)$ and z_j is sampled randomly in the feasible space.

Step 1.2. The start temperature and the end temperature are noted as T_{max} and T_f respectively, let the iteration number $k = 0$ and let $T_k = T_{max}$.

Step 2. For the $k-th$ sub-swarm, $(k = 1, 2, \cdots, n_s)$, each particle is assigned a non-domination rank ND_l and a crowding value CD_l in f space. Then, all resulting sub-swarms are combined into one population which is named as the P_t. Afterwards, each particle is assigned a non-domination rank ND_u and a crowding value CD_u in F space.

Step 3. The non-domination particles assigned both $ND_u = 1$ and $CD_u = 1$ from P_t are saved in the elite set A_t.

Step 4. For the $k-th$ sub-swarm $(k = 1, 2, \cdots, n_s)$, update the lower level decision variables.

Step 4.1. Initialize the lower level loop counter $t_l = 0$.

Step 4.2. Update the $j-th$ $(j = 1, 2, \cdots, N_l)$ particle position with the fixed upper level decision variable using Algorithm 1.

Step 4.3. $t_l = t_l + 1$.

Step 4.4. If $t_l \geqslant T_l$, go to Step 4.5. Otherwise, go to Step 4.2.

Step 4.5. Each particle of the $i-th$ sub-swarm is reassigned a non-domination rank ND_l and a crowding value CD_l in f space. Then, all resulting sub-swarms are combined into one population which is renamed as the Q_t. Afterwards, each particle is reassigned a non-domination rank ND_u and a crowding value CD_u in F space.

Step 5. Combined population P_t and Q_t to form R_t. The combined population R_t is reassigned a non-domination rank ND_u, and the particles within an identical non-domination rank are assigned a crowding distance value CD_u in the F space.

Step 6. Choose half particles from R_t. The particles of rank $ND_u = 1$ are considered first. From the particles of rank $ND_u = 1$, the particles with $CD_l = 1$ are noted one by one in the order of reducing crowding distance CD_u, for each such particle the corresponding sub-swarm from its source population (either P_t or Q_t) is copied in an intermediate population S_t. If a sub-swarm is already copied in S_t and a future particle from the same sub-swarm is found to have $ND_u = CD_u = 1$, the sub-swarm is not copied again. When all particles of $ND_u = 1$ are considered, a similar consideration is continued with $ND_u = 1$ and so on till exactly n_s sub-swarms are copied in S_t.

Step 7. Update the elite set S_t. The non-domination particles assigned both $ND_u = 1$ and $ND_l = 1$ from S_t are saved in the elite set A_t .

Step 8. Update the upper level decision variables in S_t.

Step 8.1. Based on the non-domination rank and crowding value, choose $x_u^i \in A_t$ and $x_u^j \in S_t$ using tournament selection method. Though the parents $x_u^i \in A_t$ and $x_u^j \in S_t$, the offspring $\overline{x_u^j}$ is created by the simulated binary crossover and polynomial mutation operator.

Step 8.2. For the updated $\overline{x_u^i}$, solve $\min\limits_{x_u^k \in A_t} \left|\overline{x_u^i} - x_u^k\right|$. If $x_u^p \in A_t$ meet the condition and the x_l^p is the lower lever variable corresponding to x_u^p, $\overline{x_u^i}$ and x_l^p are combined as a particle $(\overline{x_u^j}, x_l^p)$. The other $N_l - 1$ level particles are produced randomly in the feasible region. According to the above method, the n_s sub-swarms are produced.

Step 8.3. Combine all the sub-swarms produced by Step 8.2 as P_t.

Step 8.4. Every member is then assigned a non-domination rank ND_u and a crowding distance value CD_u in F space.

Step 9. $k = k + 1$, computing.

Step 10. If $T_k < T_f$ output the elite set A_t. Otherwise, go to Step 2.

In Step 4.2, denotes the non-dominated set in which the members are assigned $ND_l = 1$. In Step 9, the function for updating temperature which was adopted in Yang and Gu [40]. A relatively simple scheme is used to handle constraints. Whenever two individuals are compared, their constraints are checked. If both are feasible, non-domination sorting technology is directly applied to decide which one is selected. If one is feasible and the other is infeasible, the feasible dominates. If both are infeasible, then the one with the lowest amount of constraint violation dominates the other. Notations used in the proposed algorithm are detailed in Table 1.

Table 1: The notations of the algorithm

x_i	The $i-th$ particles position of the upper level problem.
y_j	The $j-th$ particles position of the lower level problem.
z_j	The $j-th$ particles position of BLMPP.
N_u	The population size of the upper level problem.
N_l	The sub-swarm size of the lower level problem.
t_l	Current iteration number for the lower level problem.
T_l	The pre-defined max iteration number for t_l.
ND_u	Non-domination sorting rank of the upper level problem.
CD_u	Crowding distance value of the upper level problem.
ND_l	Non-domination sorting rank of the lower level problem.
CD_u	Crowding distance value of the lower level problem.
P_t	The $t-th$ iteration population.
Q_t	The offspring of P_t.
S_t	Intermediate population.

4. Numerical Experiments

In this section, six examples will be considered to illustrate the feasibility of the proposed algorithm for problem (2.1). In order to evaluate the closeness between the obtained Pareto optimal front and the theoretical Pareto optimal front, as well as the diversity of the obtained Pareto optimal solutions along the theoretical Pareto optimal front, we adopted the following evaluation metrics:

4.1. Performance evaluation metrics

(a) Generational Distance (*GD*): This metric used by Deb [8] is employed in this paper as a way of evaluating the closeness between the obtained Pareto optimal front and the theoretical Pareto optimal

front. *GD* metric denotes the average distance between the obtained Pareto optimal front and the theoretical Pareto optimal front:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n},$$

where it is the number of the obtained Pareto optimal solutions by the proposed algorithm and is the Euclidean distance between each obtained Pareto optimal solution and the nearest member of the theoretical Pareto optimal set.

(b) Spacing (*SP*): This metric is used to evaluate the diversity of the obtained Pareto optimal solutions by comparing the uniform distribution and the deviation of solutions as described by Deb [8]:

$$SP = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{n} (\overline{d} - d_i)^2}{\sum_{m=1}^{M} d_m^e + n\overline{d}},$$

where $d_i = min_j \left(\left| F_1^i(x,y) - F_1^j(x,y) \right| + \left| F_2^i(x,y) - F_2^j(x,y) \right| \right)$ $(i, j = 1, 2, \cdots, n)$, $\overline{d}$ is the mean of all d_i, d_m^e is the Euclidean distance between the extreme solutions in obtained Pareto optimal solution set and the theoretical Pareto optimal solution set on the $m-th$ objective, M is the number of the upper level objective function, n is the number of the obtained solutions by the proposed algorithm.

4.2. Experimental comparison

All results presented in this paper have been obtained on a personal computer (CPU: AMD 2.80GHz; RAM: 3.25GB) using a $C^{\#}$ implementation of the proposed algorithm.

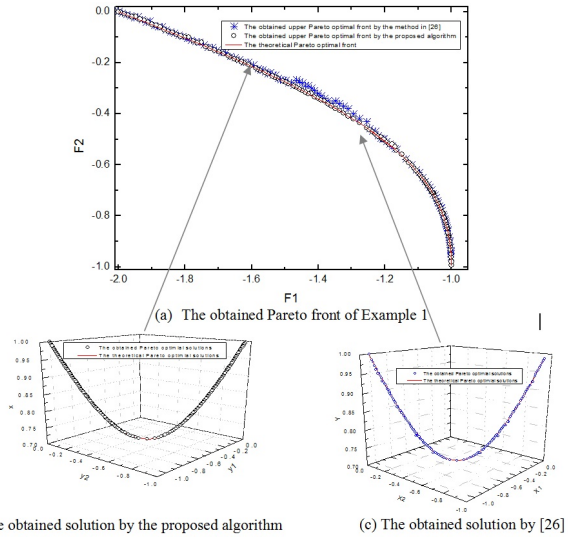
Example 4.1 ([11])**.** Let $x \in R^1$, $y \in R^2$ and set the parameters: $N_u = 200$, $T_u = 200$, $N_l = 40$, $T_l = 40$, $T_{max} = 10^6$, $T_f = 10^{-3}$ and $m = 3$.

$$\min_{x,y} F(x,y) = (y_1 - x, y_2)$$

$$s.t. G_1(y) = 1 + y_1 + y_2 \geqslant 0$$

$$\min_{y} f(x,y) = (y_1, y_2)$$

$$s.t. g_1(x,y) = x^2 - y_1^2 - y_2^2 \geqslant 0, -1 \leq y_1, y_2 \leq 1, 0 \leq x \leq 1.$$



(a) The obtained Pareto front of Example 1

(b) The obtained solution by the proposed algorithm

(c) The obtained solution by [26]

Figure 1: The obtained Pareto front and solutions of Example 4.1

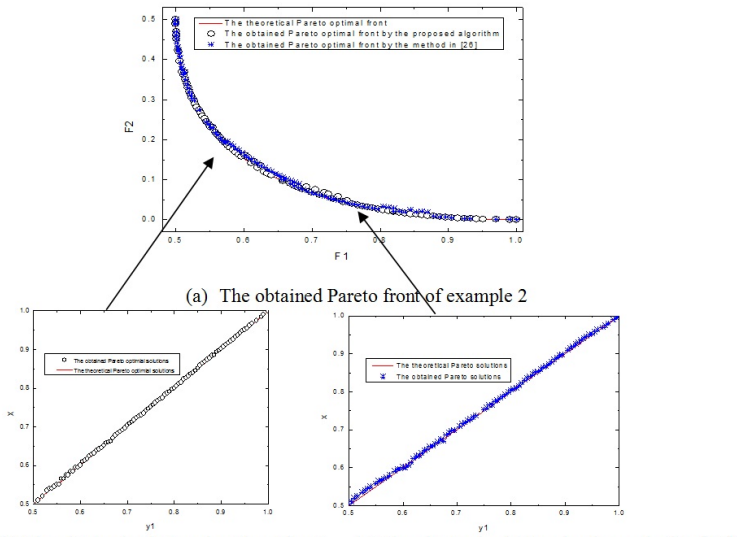
Figure 1(a) shows the obtained Pareto front of Example 4.1 by the proposed algorithm (Algorithm 2) and the method in [11]. Table 2 shows the comparison results between the two algorithms considering the metrics previously described. It can be seen that the performance of the proposed algorithm is better with respect to the generational distance, although it places slightly below the method in [11] with respect to spacing. By looking at the obtained Pareto fronts, some non-dominated vectors produced by the method in [11] are not part of the true Pareto front of the problem, however, the proposed algorithm is able to cover the full Pareto front. Figure 2(b) shows that all the obtained solutions by the proposed algorithm, which follow the relationship: $y_1 = -1 - y_2$, $y_2 = -\frac{1}{2} \pm \frac{1}{4}\sqrt{8x^2 - 4}$ and $x \in (\frac{1}{\sqrt{2}}, 1)$. However, some solutions obtained by the method in [11] do not meet the relationship.

Example 4.2. Let $x \in R^1$, $y \in R^2$ and set the parameters: $N_u = 200$, $T_u = 50$, $N_l = 40$, $T_l = 20$, $T_{max} = 10^7$, $T_f = 10^{-3}$ and $m = 3$.

$\min\limits_{x,y} F(x, y) = (x^2 + (y_1 - 1)^2 + y_2^2)$

$\min\limits_{y} f(x, y) = (y_1, y_2)$

$s.t. - 1 \leq x, y_1, y_2 \leq 2.$



(a) The obtained Pareto front of example 2

(b) The obtained solutions by Algorithm 2 (c) The obtained solutions by the method in [26]

Figure 2: The obtained Pareto front and solutions of Example 4.2

Figure 2(a) shows the obtained Pareto front of Example 4.2 by the proposed algorithm (Algorithm 2) and the method in [11]. Obviously, both the proposed algorithm and the method in [11] almost have the same spacing. However, there are some areas of the Pareto optimal solution obtained by the method in [11] that are sparse. In Figure 2(b) and Figure 2(c), it can be seen that the obtained solutions by the proposed algorithm almost follow the relationship: $x = y_1 (y_1 \in [0.5, 1])$, $y_2 = 0$. However, some areas of the solutions obtained by the method in [11] are sparse and some solutions do not meet the relationship.

Table 2: Results of Generation Distance (GD) and Spacing (SP) metrics for Examples 4.1 and 4.2

	GD		*SP*	
Test examples	The method in [11]	Algorithm 2	The method in [11]	Algorithm 2
Example 4.1	0.00266	0.00103	0.01382	0.01321
Example 4.2	0.01157	0.00995	0.00241	0.01328

Example 4.3 ([13]). Let $x \in R^{10}$, $y \in R^{10}$ and set the parameters: $N_u = 400$, $T_u = 50$, $N_l = 40$, $T_l = 20$, $T_{max} = 10^9$, $T_f = 10^{-5}$ and $m = 3$.

$$\min_{x,y} F(x,y) = ((1 + r - cos(\alpha\pi x_1)) + \sum_{j=2}^{K}(x_j - \frac{j-1}{2})^2) + \tau\sum_{j=2}^{K}(y_j - x_i)^2 - \gamma cos(\gamma\frac{\pi x_1}{2y_1}),$$

$$(1 + r - sin(\alpha\pi x_1)) + \sum_{j=2}^{K}(x_j - \frac{j-1}{2})^2) + \tau\sum_{j=2}^{K}(y_j - x_i)^2 - \gamma sin(\gamma\frac{\pi x_1}{2y_1}))$$

$$\min_{y} f(x,y) = (y_1^2 + \sum_{j=2}^{K}(y_i - x_i)^2 + \sum_{j=2}^{K} 10(1 - cos(\frac{\pi}{k}(y_i - x_i))),$$

$$y_1^2 + \sum_{j=2}^{K}(y_i - x_i)^2 + \sum_{j=2}^{K} 10\left|1 - cos(\frac{\pi}{k}(y_i - x_i))\right|)$$

$$s.t. - K \le y_i \le K, (i = 1, 2, \cdots, K); 1 \le x_1 \le 4, -K \le x_j \le K (j = 2, 3, \cdots, K)$$

$$\alpha = 1, r = 0.1, \tau = 0.1, \gamma = 1, K = 10$$

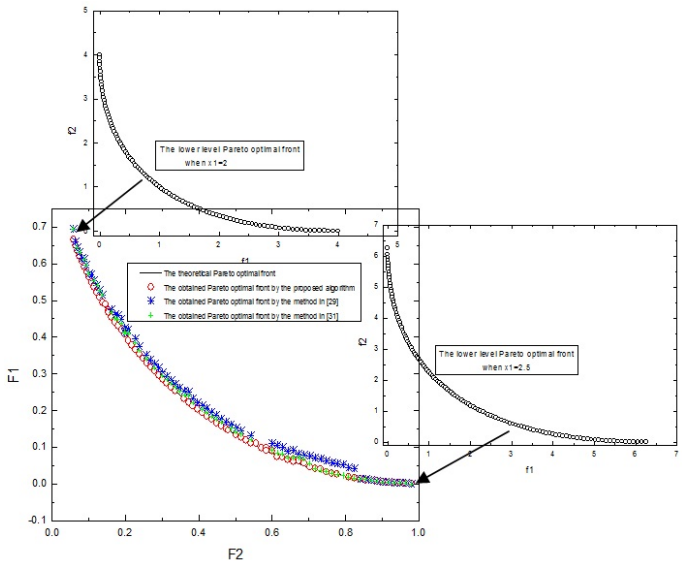


Figure 3: The obtained Pareto front of Example 4.3

This problem is more difficult than the previous problems (Example 4.1 and Example 4.2) because the lower level problem of the example has multimodalities, thereby making the lower level problem difficult in finding the upper level Pareto optimal front. Figure 3 shows the graphical results produced by the method in [13], the method in [42] and the proposed algorithm (Algorithm 2). Table 3 and Table 4 show the comparison of results among the three algorithms considering the metrics previously described. It can be seen that the performance of the proposed algorithm is the best with respect to the generational distance. By looking at the Pareto fronts of this test problem, some non-dominated vectors produced by the method in [13] are not part of the true Pareto front and there are some areas of the Pareto optimal solution obtained by the method in [42] that are sparse. However, the proposed algorithm is able to cover the full Pareto optimal front. Although it places slightly below the method in [13] with respect to spacing, the fact that the spacing metric provides a good value is irrelevant if the non-dominated vectors produced by the algorithm are not part of the true Pareto front of the problem. Furthermore, two obtained lower level Pareto optimal fronts are given when $x_1 = 2$ and $x_1 = 2.5$.

Example 4.4 ([13]). Let $x \in R^{10}$, $y \in R^{10}$ and set the parameters: $N_u = 400$, $T_u = 50$, $N_l = 40$, $T_l = 20$, $T_{max} = 10^{11}$, $T_f = 10^{-5}$ and $m = 3$.

$$\min_{x,y} F(x,y) = v_1(x_1) + \sum_{j=2}^{K} y_j^2 + 10(1 - cos(\frac{\pi}{k})y_i) + \tau \sum_{i=2}^{K}(y_i - x_i)^2 - rcos(\gamma \frac{\pi x_1}{2y_1}),$$

$$v_2(x_1) + \sum_{j=2}^{K} y_j^2 + 10(1 - cos(\frac{\pi}{k})y_i) + \tau \sum_{i=2}^{K}(y_i - x_i)^2 - rsin(\gamma \frac{\pi x_1}{2y_1})$$

$$\min_{y} f(x,y) = (y_1^2 + \sum_{j=2}^{K}(y_i - x_i)^2, i\sum_{i=2}^{K}(y_i - x_i)^2)$$

$$s.t. - K \le y_i \le K, (i = 1, 2, \cdots, K); 0.001 \le x_1 \le 4, -K \le x_j \le K (j = 2, 3, \cdots, K).$$

$$\alpha = 1, r = 0.1, \tau = 0.1, \gamma = 1, K = 10.$$

where

$$v_1(x_1) = \begin{cases} cos(0.2\pi)x_1 + sin(0.2\pi)\sqrt{|0.02sin(5\pi x_1)|} & for \quad 0 \le x_1 \le 1 \\ x_1 - (1 - sin(0.2\pi)) & for \quad x_1 \ge 1 \end{cases}$$

$$v_2(x_1) = \begin{cases} -sin(0.2\pi)x_1 + cos(0.2\pi)\sqrt{|0.02sin(5\pi x_1)|} & for \quad 0 \le x_1 \le 1 \\ 0.1(x_1 - 1) - sin(0.2\pi)) & for \quad x_1 \ge 1 \end{cases}$$

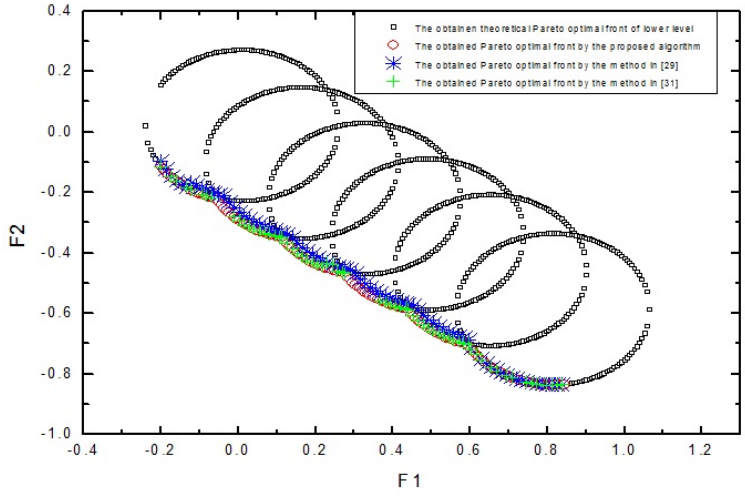


Figure 4: The obtained Pareto front of Example 4.4

For the Example 4.4, the upper level problem has multimodalities, thereby causing an algorithm difficulty in finding the upper level Pareto optimal front. Figure 4 shows the obtained Pareto front of Example 4.4 by the method in [13], the method in [42] and the proposed algorithm (Algorithm 2). Table 3 and Table 4 show the comparison of results among the three algorithms considering the metrics previously described. It can be seen that the performance of the proposed algorithm is better than the method in [13] with respect to the generational distance though they almost have the same performance with respect to the spacing. On the other hand, the method in [42] and the proposed algorithm almost have the same generational distance, but the proposed algorithm is better with respect to spacing. It shows that the global search ability of the proposed algorithm is better than the method in [42]. Moreover, all corresponding lower level Pareto optimal fronts are given.

Example 4.5 ([13]). Let $x \in R^{10}$, $y \in R^{10}$ and set the parameters: $N_u = 400$, $T_u = 50$, $N_l = 40$, $T_l = 20$, $T_{max} = 10^{13}$, $T_f = 10^{-5}$ and $m = 3$.

$$\min_{x,y} F(x,y) = (x_1 + \sum_{j=3}^{K}(x_j - j/2)^2 + \tau\sum_{i=3}^{K}(y_i - x_i)^2 - cos(4tan^{-1}(\frac{x_2 - y_2}{x_1 - y_1})),$$

$$x_2 + \sum_{j=3}^{K}(x_j - j/2)^2 + \tau\sum_{i=3}^{K}(y_i - x_i)^2 - cos(4tan^{-1}(\frac{x_2 - y_2}{x_1 - y_1})))$$

$$s.t.\, G(x) = x_2 - (1 - x_1^2)^2 \geq 1$$

$$\min_{y} f(x,y) = (y_1 + \sum_{i=3}^{K}(y_i - x_i)^2, y_2 + \sum_{i=3}^{K}(y_i - x_i)^2)$$

$$s.t.\, g_1(x,y) = (y_1 - x_1)^2 + (y_2 - x_2)^2 \leq r^2$$

$$-K \leq y_i \leq K, (i = 1, 2, \cdots, K)$$

$$0 \leq x_1 \leq K, -K \leq x_j \leq K (j = 2, 3, \cdots, K).$$

$$\tau = 1, r = 0.2, K = 10.$$

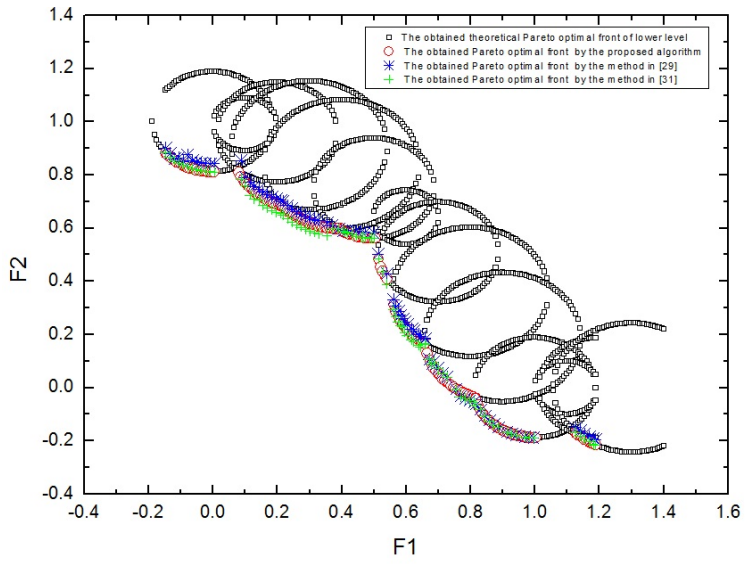


Figure 5: The obtained Pareto front of Example 4.5

Figure 5 shows the obtained Pareto front of Example 4.5 by the method in [13], the method in [42] and the proposed algorithm (Algorithm 2). Table 3 and Table 4 show the comparison of results among the three algorithms considering the metrics previously described. For this example, the graphical results again indicate that the method in [13] does not cover the full Pareto front. Since the non-dominated vectors found by the method in [13] are clustered together, the spacing metric provides very good results. The method in [42], some non-dominated vectors produced are relatively sparse in some regions of the true Pareto optimal front and some non-dominated vectors produced are slightly off the true Pareto front. Graphically, it can be seen the proposed algorithm is able to cover the entire Pareto front. It is also interesting to note that the Pareto optimal fronts for both the lower and upper level lie on constraint boundaries and every lower level Pareto optimal front has an unequal contribution to the upper level Pareto optimal front.

Example 4.6 ([13]). Let $x \in R^1$, $y \in R^9$ and set the parameters: $N_u = 400$, $T_u = 50$, $N_l = 40$, $T_l = 20$, $T_{max} = 10^7$, $T_f = 10^{-3}$ and $m = 3$.

$$\min_{x,y} F(x,y) = ((1-y_1)(1+\sum_{j=2}^{K} x_1), y_1(1+\sum_{j=2}^{K} y_j^2)x_1)$$

$$s.t.\, G(x) = -(1 - y_1x_1 - \frac{1}{2}x_1y_1) \leq 1$$

$$\min_{y} f(x,y) = ((1-y_1)(1+\sum_{K+1}^{K+L} y_j^2)x_1, y_1(1+\sum_{K+1}^{K+L} y_j^2)x_1)$$

$$s.t.\, 1 \leq x_1 \leq 2, -1 \leq y_1 \leq 1$$

$$-(K+L) \leq y_j \leq (K+L)(j = 2, 3, \cdots, K+L), K = 5, L = 4.$$

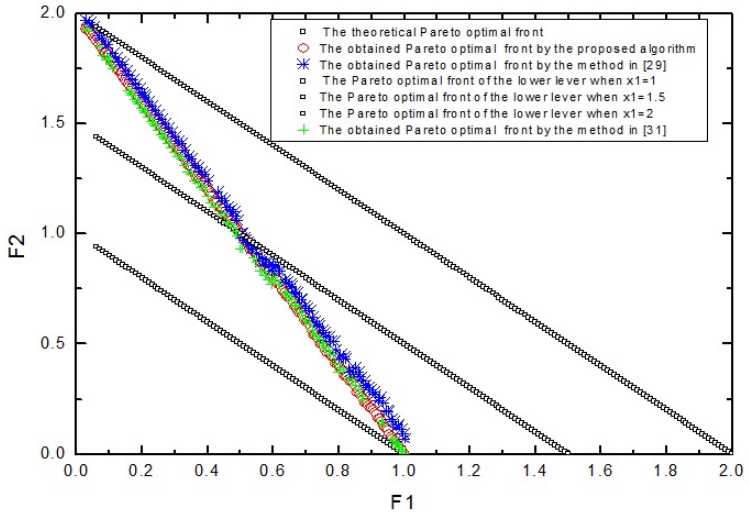


Figure 6: The obtained Pareto front of Example 4.6

Figure 6 shows the obtained Pareto front of Example 4.6 by the method in [13], the method in [42] and the proposed algorithm (Algorithm 2). Table 3 and Table 4 show the comparison of results among the three algorithms considering the metrics previously described. It can be seen that our algorithm and the method in [13] almost have the same performance of the spacing, but some non-dominated vectors produced by the method in [13] are slightly off the true Pareto front. On the other hand, the method in [42] and the proposed algorithm almost have the same generational distance. However, some areas of the Pareto optimal solution obtained by the method in [42] are sparse. Moreover, three obtained lower level Pareto optimal fronts are given when $y_1 = 1$, $y_1 = 1.5$ and $y_1 = 2$. It can be seen that only one Pareto optimal point from each participating lower level problem qualifies to be on the upper level Pareto optimal front.

Table 3: Results of the Generation Distance (GD) metrics for the above four examples

Test examples	The proposed algorithm	The method in [42]	The method in [13]
Example 4.3	0.00021	0.00127	0.00832
Example 4.4	0.00029	0.00089	0.06391
Example 4.5	0.00053	0.00375	0.02574
Example 4.6	0.00019	0.00098	0.00769

5. Application of the Proposed Algorithm for a Watershed Water Trading Decision-Making Problem

Taking into account water rights and emission rights to establish a water market system based on watershed management is the most effective economic method to solve water shortage and pollution of watershed In this water market system, the watershed management agency usually to maximize his profits, whereas each user goal is to maximize its own profit. The problem involves uncertainty and is bilevel in nature, thus, a watershed water trading decision-making model based on bilevel programming is constructed, in which the upper decision-maker is the watershed management agency as the planning, controlling and coordinating center of watershed and each user is the lower decision-maker. In the model, the index of water pollution is added to the target function of watershed management agency based on the [38], setting maximum benefit and minimum water pollution as the upper objective; while the maximum benefit of different water users as the lower objective. The bilevel multiobjective programming model is built as following:

$$\begin{aligned}
&\max_{w,t,r_1,g_1,r_2,g_2} V_T = 0.4w + t(q_1+q_2) + f_1 + f_2\\
&\max_{q_1,q_2} -S = -100q_1 - 300q_2\\
&s.t.\, r_1 + r_2 + w = 90,\\
&\quad q_1 + q_2 + w \le 90,\\
&\quad g_1 + g_2 = 20\\
&\quad r_1 \ge 38, r_2 \ge 42, g_1 \ge 7, g_2 \ge 8, w \ge 6, 0.3 \le t \le 2\\
&\max_{q_1,l_1} V_1 = 0.7q_1 - q_1 t - 0.3(45-q_1)^2 + (r_1-q_1)[0.9 - 0.01(r_1+r_2-q_1-q_2)]\\
&\qquad\qquad -0.2(0.2q_1 - l_1)^2 + (g_1-l_1)[0.8 - 0.01(g_1+g_2-l_1-l_2)]\\
&\max_{q_2,l_2} V_2 = 0.8q_2 - q_2 t - 0.2(47-q_2)^2 + (r_2-q_2)[0.9 - 0.01(r_1+r_2-q_1-q_2)]\\
&\qquad\qquad -0.1(0.3q_2 - l_2)^2 + (g_2-l_2)[0.8 - 0.01(g_1+g_2-l_1-l_2)]\\
&s.t.\, l_1 + l_2 \le 20,\\
&\quad q_1 \ge 0, l_1 \ge 0\\
&\quad q_2 \ge 0, l_2 \ge 0.
\end{aligned}$$

Where q_1 and q_2 are actual water intake volume of water consumer A and water consumer B, respectively. l_1 and l_2 are waste water discharge volume of two users respectively. r_1 and r_2 are water rights of two users respectively. g_1 and g_2 are emission rights of two users respectively. w is ecological water requirement of watershed. t is water resource fees. Here $f_1 = (q_1) = 0.7q_1$, $f_2 = (q_2) = 0.8q_2$. The BOD of water consumer A is $1.0 \times 10^5 kg/billion(m^3)$ and the BOD of water consumer B is $1.5 \times 10^5 kg/billion(m^3)$.

The optimal solution of the problem is obtained by the Algorithm 2 (The parameter is set as: $T_u = 10$, $N_l = 10$, $T_i = 20$, $T = 40$). We execute the algorithm in 20 independent runs and the Pareto optimal solution is obtained. The solution is as following: $q_1 = q_2 = 4.2 \times 10^9(m^3)$, $l_1 = 7.0216 \times 10^8(m^3)$,

Table 4: Results of the Spacing (SP) metrics for the above four examples

Test examples	The proposed algorithm	The method in [42]	The method in [13]
Example 4.3	0.00125	0.01921	0.00097
Example 4.4	0.00235	0.00986	0.00241
Example 4.5	0.00371	0.00793	0.02082
Example 4.6	0.00150	0.00205	0.00149

$l_2 = 9.0013 \times 10^8(m^3)$, $r_1 = 4 \times 10^9(m^3)$, $r_2 = 4.2 \times 10^9(m^3)$, $g_1 = 9 \times 10^8(m^3)$, $g_2 = 1.1 \times 10^9(m^3)$, $w = 6 \times 10^9(m^3)$, $t = 0.3yuan/m^3$, $V_T = 5.90613 \times 10^9 yuan$, $-S = -1.68 \times 10^3 kg$, $V_1 = 1.34240 \times 10^9 yuan$, $V_2 = 1.80244 \times 10^9 yuan$.

6. Conclusion

In this paper, an improved SA algorithm is presented, in which a heuristic criterion for determining the temperature updating function of SA algorithm is applied, enabling the particle to escape the local optima. The improved SA algorithm is employed for solving BLMPP for the first time. In this study, some numerical examples are used to explore the feasibility and efficiency of the proposed algorithm. The experimental results indicate that the obtained Pareto front by the proposed algorithm is very close to the theoretical Pareto optimal front, and the solutions are also distributed uniformly on entire range of the theoretical Pareto optimal front. The proposed algorithm is simple and easy to implement, which provides another appealing method for further study on BLMPP.

Acknowledgements

The authors would like to gratefully thank the associated editor and the three anonymous referees for their valuable comments and suggestions, which have helped to improve the paper. This work is supported by the National Science Foundation of China (61273179, 11401487, 11201039 and 11501233), the Young Project of Hubei Provincial Department of Education (Q20141304), the Dr. Start-up fund by the Yangtze University (2014), the China Postdoctoral Science Foundation(2015M582512) and the Fundamental Research Funds for the Central Universities (XDJK2014C073).

References

[1] M. Abo-Sinna, I. Baky, *Interactive balance spaces approach for solving multi-level multi-objective programming problems*, Inform. Sci., **177** (2007), 3397–3410. 1
[2] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, *A simulated annealing based multi-objective optimization algorithm: AMOSA,* IEEE Trans. Evolutionary Comput., **12** (2008), 269–283. 1
[3] J. F. Bard, *Practical bilevel optimization: Algorithms and applications,* Kluwer Academic, Dordrecht, (1998). 1
[4] J. Chen, Q. H. Ansari, Y. C. Liou, J. C. Yao, *A proximal point algorithm based on decomposition method for cone constrained multiobjective optimization problems,* Comput. Opti. Appl., **2016** (2016), 20 pages.
[5] J. Chen, Y. C. Liou, C. F. Wen, *Bilevel vector pesudomonotone equilibrium problem: duality and existence,* J. Nonlinear Convex Anal., **16** (2015), 1293–1303.
[6] B. Colson, P. Marcotte, G. Savard, *Bilevel programming: A survey,* A Quarterly J. Oper. Res., **3** (2005), 87–107.
[7] B. Colson, P. Marcotte, G. Savard, *An overview of bilevel optimization,* Ann. Oper. Res., **153** (2007), 235–256. 1
[8] K. Deb, *Multi-Objective Optimization using evolutionary algorithms,* IEEE Trans. Evolutionary Comput., **6** (2002), 182–197. 4.1
[9] K. Deb, S. Agrawalan, A. Pratap, T. Meyarivan, *A fast and elitist multi-objective genetic algorithm: NSGA-II.*, IEEE Trans. Evolutionary Comput., **6** (2002), 182–197. 3.2
[10] K. Deb, A. Sinha, *Constructing test problems for bilevel evolutionary multi-objective optimization,* IEEE Congr. Evolutionary Comput., (2009), 1153–1160. 1
[11] K. Deb, A. Sinha, *Solving bilevel multiobjective optimization problems using evolutionary algorithms,* Lecture notes in computer science, **5467** (2009), 110–124. 4.1, 4.2, 4.2, 2
[12] K. Deb, A. Sinha, *An evolutionary approach for bilevel multiobjective problems,* Commun. Comput. Inform. Sci., **35** (2009), 17–24. 1
[13] K. Deb, A. Sinha, *An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary local-search algorithm,* Evolutionary Comput., **18** (2010), 403–449. 1, 4.3, 4.2, 4.4, 4.2, 4.5, 4.2, 4.6, 4.2, 3, 4
[14] S. Dempe, *Foundations of bilevel programming, Non-convex optimization and its applications series,* Kluwer Academic, Dordrecht, (2002). 1
[15] S. Dempe, *Annotated bibliography on bilevel programming and mathematical problems with equilibrium constraints,* Optim., **52** (2003), 333–359. 1
[16] G. Eichfelder, *Solving nonlinear multi-objective bi-level optimization problems with coupled upper level constraints,* Technical Report Preprint No. 320, Preprint-Series of the Institute of Applied Mathematics, Univ. Erlangen-Nrnberg, Germany, (2007). 1

[17] G. Eichfelder, *Multiobjective bilevel optimization,* Math. Program., **123** (2010), 419–449.1
[18] Y. Gao, G. Zhang, J. Lu, H. Wee, *Particle swarm optimization for bilevel pricing problems in supply chains,* J. Global Optim., **51** (2011), 245–254.1
[19] H. S. Kemal, A. R. Ciric, *A dual temperature simulated annealing approach for solving bilevel programming problems,* Comput. Chem. Eng., **23** (1998), 11–25.1
[20] S. Kirkpatrick, C. Gelatt, M. Vecchi, *Optimization by simulated annealing,* Science, **220** (1983), 671–680.1
[21] R. J. Kuo, C. C. Huang, *Application of particle swarm optimization algorithm for solving bilevel linear programming problem,* Comput. Math. Appl., **58** (2009), 678–685.1
[22] Z. Q. Luo, J. S. Pang, D. Ralph, *Mathematical programs with equilibrium constraints,* Cambridge University Press, Cambridge, (1996).1
[23] F. Maffioli, *Randomized heuristic for NP-hard problem,* Advanced School on Stochastic in Combinatorial Optimization, World Sci. publishing, Singapore, (1987), 76–93.1
[24] I. Nishizaki, M. Sakawa, *Stackelberg solutions to multiobjective two-level linear programming problems,* J. Optim. Theory Appl., **103** (1999), 161–182.1
[25] V. Oduguwa, R. Roy, *Bilevel optimization using genetic algorithm,* Proc. 2002 IEEE International Conference on Artificial Intelligence Systems, (2002), 322–327.1
[26] X. Shi, H. Xia, *Interactive bilevel multi-objective decision making,* J. Oper. Res. Soc., **48** (1997), 943–949.1
[27] X. Shi, H. Xia, *Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers,* J. Multi-criteria Decision Anal., **10** (2001), 27–34.1
[28] K. Shimizu, Y. Ishizuka, J. F. Bard, *Non-differentiable and two-level mathematical programming,* Kluwer Academic, Dordrecht, (1997).1
[29] A. Sinha, *Bilevel multi-objective optimization problem solving using progressively interactive EMO,* Proceeding of the Sixth International Conference on Evolutionary Multi-criterion Optimization, (2011).1
[30] A. Sinha, K. Deb, *Towards understanding evolutionary bilevel multiobjective optimization algorithm,* Technical Report Kangal Report No. 2008006, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India, (2008).1
[31] K. Smith, R. Everson, J. Fieldsend, *Dominance measures for multi-objective simulated annealing,* Proc. 2004 IEEE Congr. Evolutionary Comput. (CEC'04), (2004), 23–30.1, 3.1
[32] B. Suman, P. Kumar, *A survey of simulated annealing as a tool for single and multiobjective optimization*, J. Oper. Res. Soc., **57** (2006), 1143–1160.1
[33] L. N. Vicente, P. H. Calamai, *Bilevel and multilevel programming: A bibliography review,* J. Global Optim., **5** (1994), 1–23.1
[34] L. N. Vicente, P. H. Calamai, *Bilevel and multilevel programming: A bibliography review*, J. Global Optim., **5** (2004), 291–306.1
[35] Y. Wang, Y. C. Jiao, H. Li, *An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme,* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, **35** (2005), 221–232.1
[36] G. Wang, Z. Wan, X. Wang, *Bibliography on bilevel programming,* Adv. Math. (in Chinese), **36** (2007), 513–529.1
[37] U. P. Wen, S. T. Hsu, *Linear bilevel programming problems-A review,* J. Oper. Res. Soc., **42** (1991), 125–133.1
[38] C. Y. Wu, Y. Z. Zhao, *Watershed water trading decision-making model based on bilevel programming*, Oper. Res. Management Sci. (in Chinese), **20** (2011), 30–37.5
[39] R. Xiang, G. Y. Lu, D. He, *Simulated annealing algorithm for solving a bi-level optimization model on high-speed railway station location,* 2010 Third International Conference on Inform. Comput., (2010), 159–162.1
[40] R. L. Yang, J. F. Gu, *An efficient simulated annealing algorithm for global optimization,* Syst. Eng.-Theory Practice (in Chinese), **17** (1997), 30–36.3.1, 3.2
[41] Y. Yin, *Genetic algorithm based approach for bilevel programming models,* J. Trans. Eng., **126** (2000), 115–120.1
[42] T. Zhang, T. S. Hu, X. N. Guo, Z. Cheng, Y. Zheng, *Solving high dimensional bilevel multiobjective programming problem using a hybrid particle swarm optimization algorithm with crossover operator,* Knowledge-Based Sys., **53** (2013), 13–19.1, 4.2, 4.2, 4.2, 4.2, 3, 4
[43] G. Zhang, J. Lu, *Fuzzy bilevel programming with multiple objectives and cooperative multiple followers,* J. Global Optim., **47** (2010), 403–419.1