Research Article

The Artificial Neural Networks Based on Scalarization Method for a Class of Bilevel Biobjective Programming Problem

Tao Zhang,1 Zhong Chen,1 June Liu,2 and Xiong Li3

1School of Information and Mathematics, Yangtze University, Jingzhou 434023, China
2School of Management, Huaibei Normal University, Huaibei 235000, China
3School of Mathematical Sciences, Beijing Normal University, Beijing 100875, China

Correspondence should be addressed to Tao Zhang; zt_math981@126.com

Received 6 January 2017; Revised 19 May 2017; Accepted 7 August 2017; Published 14 September 2017

A two-stage artificial neural network (ANN) based on scalarization method is proposed for bilevel biobjective programming problem (BLBOP). The induced set of the BLBOP is firstly expressed as the set of minimal solutions of a biobjective optimization problem by using scalar approach, and then the whole efficient set of the BLBOP is derived by the proposed two-stage ANN for exploring the induced set. In order to illustrate the proposed method, seven numerical examples are tested and compared with results in the classical literature. Finally, a practical problem is solved by the proposed algorithm.

1. Introduction

The bilevel programming problem (BLP) is a nested optimization problem with two levels in a hierarchy: the upper and lower level decision-makers. The upper level maker makes his decision firstly, followed by the lower level decision-maker. The objective function and constraint of the upper level problem not only rely on their own decision variables but also depend on the optimal solution of the lower level problem. The decision-maker at the lower level has to optimize his own objective function under the given parameters from the upper level decision-maker. Since many practical problems, such as engineering design, management, economic policy, and traffic problems, can be formulated as hierarchical problems, BLP has been studied and received increasing attention in the literatures. During the past decades, some surveys and bibliographic reviews were given by several authors [1–4]. Reference books on bilevel programming and related issues have emerged [5–8].

The bilevel programming problem is a nonconvex problem, which is extremely difficult to solve. As we know, BLP is a NP-Hard problem [9–11]. Vicente et al. [12] also showed that even the search for the local optima to the bilevel linear programming is NP-Hard. Even so, many researchers are devoted to develop the algorithms for solving BLP and propose many efficient algorithms. To date a few algorithms exist to solve BLP; it can be classified into four types: Karush-Kuhn-Tucker approach (KKT) [13–16], Branch-and-bound method [17], penalty function approach [18–21], and descent approach [22, 23].

Unfortunately, the bilevel programming problem is nonconvex and the properties such as differentiation and continuity are necessary when proposing the traditional algorithms. Thus, many researchers tend to propose the heuristic algorithms for solving BLP because of their key characteristics of minimal problem restrictions such as differentiation. Mathieu et al. [24] firstly developed a genetic algorithm (GA) for bilevel linear programming problem because of its good characteristics such as simplicity, minimal problem restrictions, global perspective, and implicit parallelism. Motivated by the same reason, other kinds of genetic algorithm for solving bilevel programming were also proposed in [25–28]. Because of the prominent advantage that neural computing can converge to the equilibrium point (optimal solution) rapidly, the neural network approach was used to solve bilevel programming problem in [29–31]. In additional, since McCulloch and Pitts [32] and Pyne [33] utilized logical calculus to emulate nervous activities, there have been various
types of analogue neural networks proposed for computation. Sheng et al. [34] firstly proposed a neural network approach based on Frank-Wolfe method for a class of BLP problems. Shih et al. [35] and Lan et al. [29] presented neural network for solving linear BLP problem, respectively. Recently, Lv et al. [30, 36] investigated the nonlinear bilevel programming and the convex quadratic bilevel programming by neural network. However, there are few results on application of neural networks to the BLMPP in the literature.

Particle swarm optimization (PSO) is a relatively novel heuristic algorithm inspired by the choreography of a bird flock. Due to its high speed of convergence and relative simplicity, the PSO algorithm has been employed for solving BLP problems. For example, Li et al. [37] proposed a hierarchical PSO for solving BLP problem. Kuo and Huang [38] applied the PSO algorithm for solving bilevel linear programming problem. Jiang et al. [39] presented the PSO based on CHKS smoothing function for solving nonlinear bilevel programming problem. Gao et al. [40] presented a method to solve bilevel pricing problems in supply chains using PSO. Zhang et al. [41] presented a new strategic bidding optimization technique which applies bilevel programming and swarm intelligence. The hybrid algorithms based on PSO are also proposed to solve the bilevel programming problems [42–44]. Besides, Tabu search [45–47], simulated annealing [48], ant colony optimization [49], and λ-cut and goal-programming-based algorithm [50] are also typical intelligent algorithms for solving bilevel programming problem.

However, the algorithms mentioned above are only for the simple single objective bilevel programming problems. In fact, the multiobjective characteristics widely existing in the BLPP and the bilevel multiobjective programming problem (BLMPP) have attracted many researchers’ interesting. For example, Shi and Xia [51, 52], Abo-Sinna and Baky [53], and Nishizaki and Sakawa [54] presented an interactive algorithm for BLMPP. Eichfelder [55] developed a numerical method for solving nonlinear nonconvex bilevel multiobjective optimization problems. In recent years, the metaheuristic has attracted considerable attention as an alternative method for BLMPP. For example, Deb and Sinha [56–58] as well as Sinha and Deb [59] discussed BLMPP based on evolutionary multiobjective optimization principles. Based on those studies, Deb and Sinha [60] proposed a viable and hybrid evolutionary-local-search based algorithm and presented challenging test problems. Sinha [61] presented a progressively interactive evolutionary multiobjective optimization method for BLMPP. Lately, Zhang et al. [62] proposed an improved PSO for BLMPP and a framework of PSO for solving BLMPP is established. Subsequently, Zhang et al. [63] proposed an elite quantum behaved PSO for relatively complex BLMPP. In 2013, Zhang et al. [64] proposed a hybrid particle swarm optimization algorithm with crossover operator to solve high-dimensional BLMPP. Almost all the research object of the BLMPPs is the BLBOP, so we mainly consider the BLBOP in this paper.

As we known, the authenticity of the lower level Pareto optimal solution is very important for the BLBOP. If the obtained optimal Pareto solutions possess the fraudulence, it can lead to the failure to solve the whole problem. In this paper, the induced set of the BLBOP is firstly expressed as the set of minimal solutions of a biojective optimization problem by a scalar approach which can greatly improve the accuracy of the lower level Pareto optimal solutions. Based on the efficient set of the BLBOP, a two-stage ANN is presented for solving whole problem which can reduce the computation burden.

The remaining of this paper is organized as follows. In Section 2, we give the formulation of the model and related definitions. In Section 3, we will introduce the scalar approach for the induced set of the BLBOP and the two-stage ANN algorithm for the whole problem. In Section 4, some numerical examples and practical problem are given to demonstrate the feasibility and efficiency of the proposed algorithm, while the conclusion is reached in Section 5.

2. Problem Formulation and Main Theory Results

Let $X$ be a nonempty subset of $\mathbb{R}^{n_1}$, $Y$ be a nonempty subset of $\mathbb{R}^{n_2}$, and $F: \mathbb{R}^{n_1} \times \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{m_1}$, $f: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{m_2}$, $G: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^p$, and $g: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^q$ be vector-valued mappings. We consider the following bilevel multiobjective programming problem (BLMPP):

\[
\begin{align*}
\min_{(x,y)} & \quad F(x, y) \\
\text{s.t.} & \quad x \in I, \\
\min_y & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \geq 0, \quad x \in X, \ y \in Y,
\end{align*}
\]

where $F(x, y)$ and $f(x, y)$ are the upper level and the lower level objective functions, respectively, $x \in I \subseteq X$ and $g(x, y)$ denote the upper level and the lower level constraints, respectively. Let $I = [c, d] \subset X$ and $G = \{(x, y) \mid g(x, y) \geq 0, \ x \in I\}$. For the fixed $x \in I$, let $G(I)$ denote the Pareto optimal solutions to the lower level problem; the induced set of problem (1) is denoted as $\Omega = \{(x, y) \mid (x, y) \in G, \ y \in G(I)\}$. Note that the constraint $x \in I$ is uncoupled from the lower level variable $y \in Y$. Particularly, if $m_1 = m_2 = 2$, we also call the BLMPP as bilevel biobjective programming problem (BLBOP). In the following, we shall focus on the BLBOP and assume that $m_1 = m_2 = 2$ and $n_1 = 1$.

Definition 1. Let $K$ be a closed pointed convex cone. A point $(\overline{x}, \overline{y}) \in G$ is called a Pareto optimal solution of the lower level problem with respect to $K$ if $(f(\overline{x}, \overline{y})-K) \cap f(G) = \{f(\overline{x}, \overline{y})\}$.

Definition 2. For a fixed $x \in I$, if $y$ is a Pareto optimal solution to the lower level problem, then $(x, y)$ is a feasible solution to problem (1).

Definition 3. If $(x^*, y^*)$ is a feasible solution to problem (1) and there are no $(x, y) \in \Omega$, such that $F(x, y) < F(x^*, y^*)$, then $(x^*, y^*)$ is a Pareto optimal solution to problem (1), where “<” denotes Pareto preference.
Definition 4. If \((x^*, y^*)\) is the optimistic solution for problem (1), then \((x^o, y^o)\) is given by \(\min_{x,y} \{F(x,y) \mid y \in G(x), x \in I\}\).

Remark 5. The optimistic solution to the BLBOP is the one that optimizes the leader's objective function over the set of efficient solutions to the follower, assuming that the follower has no preferences among the efficient solutions obtained for each leader's decision \(x\) or that the follower will choose the solution that most benefits the leader. In this paper, we only consider the optimistic BLBOP.

For problem (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 optimal solutions of the lower level problem as the optimal response feedback to the upper level problem, and this point of view is accepted usually. On the other hand, the authenticity of the lower level Pareto optimal solution is very important for the BLMP. If the obtained optimal Pareto solutions possess the fraudulence, it can lead to the failure to solve the whole problem. In this paper, we proposed the scalar approach for the lower level Pareto optimal solutions in order to improve the accuracy of the lower level Pareto optimal solutions.

3. Algorithm

3.1. The Scalar Approach for the Induced Set of BLBOP. In bilevel optimization, the constraint set of the upper level problem is given by the solution set of the lower level optimization problem. According to Theorem 4.1 of the literature [55], the induced set \(\Omega\) of problem (1) is equivalent to the Pareto optimal set of the multiobjective optimization problem

\[
\min_y f(x, y) = \left(\frac{f(x, y)}{x}\right) \quad (2)
\]

s.t. \((x, y) \in G\)
\(x \in I\).

Thus, to solve the induced set of BLBOP is transformed to solve the Pareto optimal solution set of problem (2). Inspired by the scalar approach adopted in [55], the approximation for the induced set of problem (1) can be solved by the following algorithm.

Algorithm 1.

Step 1. Discretize the upper level variable.

Step 1.1 For one-dimension upper level decision variable, choose \(\theta > 0\) and discretize \(I = [c, d]\) by \(x_1 = c, x_k = x_1 + (k - 1)\theta (k = 2, 3, \ldots, n - 1), x_n = d\).

Step 1.2 For \(n\)-dimension \((n \geq 2)\) upper level decision variable, choose \(\theta_j > 0\) and discretize \(x \in I = [c_1, d_1] \times [c_2, d_2] \times \cdots \times [c_n, d_n]\) by \(x \in R^n \mid x_i = q_i + n_i\theta_i, x_i \leq d_i\).

Step 2. Initialize the loop variables \(k = 1\) and predefined an accuracy measure \(\alpha > 0\).

Step 3. Execute the following steps for \(x = x^k (k = 1, 2, \ldots, n)\) and let \(\overline{a} = \hat{x}^k\).

Step 3.1 Solve \(f_j(x^k, \overline{y}^k) = \min_x \{f_j(x, y) \mid (x, y) \in G\}\) and \(f_l(x^k, \overline{y}^k) = \min_x \{f_l(x, y) \mid (x, y) \in G\}\). Then, determine \(a^1 = (f_j(x^k, \overline{y}^k), 0)\) and \(a^M = (f_l(x^k, \overline{y}^k), 0)\). Denote the approximate Pareto optimal solution set by \(A^0 = \{(x^k, y^k)\}\).

Let \(a^2 = (f_j(x^k, \overline{y}^k) + \delta, 0)^T\) with a small \(\delta > 0\) and \(t = 2\).

Step 3.2 If \(a^1 \leq a^M\), solve problem (2) using the scalarization method \(SP(d, r, \overline{a})\) in [56] with \(\overline{a} = x^k, r = (0, 1)^T\) for the minimal solution \(y^*\). Set \(A^0 = A^0 \cup \{ (x^k, y^k)\}\).

Step 3.3 Calculate \(M, N\) according to Theorem 4.3 and determine \(\overline{A}\) according to (5.2) in [56]. Set \(a^{t+1} = a^1 + \overline{A} (1, 0)^T\) and let \(t = t + 1\); go to Step 4.

Step 4. Output the approximation Pareto optimal solution set \(A^0\); that is, \(A^0 = \bigcup_{k=1}^{p} A^0_k\).

3.2. The Two-Stage ANN. If \(\theta > 0\) in Step 2 is small, then set \(\overline{A}\) will consist of many points. Then, the determination of all nondominated points for the upper level problem can be very expensive. In this subsection, the two-stage ANN is presented for determining the Pareto optimal solution of the BLBOP on the induced set \(\overline{A}\). The first stage is to map the vectors from \(R^{n+m}_o\) to \(R^m\) and the second stage is to determine the Pareto optimal solutions of problem (1). The detail of each stage is described as follows.

The first stage of the ANN is a feed-forward artificial neural network (FFANN) which is composed by two subnetworks \(A\) and \(B\) with the same structure and neuron output function. For each subnetwork, the nodes are organized into two layers and the weighted arcs only link nodes in lower layers to nodes in higher layers. The first stage is to compute the objective function value for the upper layer.

The second stage is a quasineural artificial network; namely, the network has no connectivity weight and the output value can be computed directly by software. The input layer of the quasineural artificial network is the output layer of the first stage network. For the hidden layer, the input of and the output are defined by (3) and (4), respectively.

\[
S_p = f_p \left( x^k_A - f_p \left( x^k_A \right) \right) \quad p = 1, 2, \quad (3)
\]

\[
O_\delta \left( S_p \right) = \begin{cases} 1 & \text{if } S_p > 0 \\ -1 & \text{if } S_p \leq 0 \end{cases} \quad p = 1, 2, \quad (4)
\]

For the output layer, the input of and the output are defined by (5) and (6), respectively.

\[
S_q = \sum_{p=1}^{2} O_\delta \left( S_p \right), \quad (5)
\]
Based on the set \( A^0 \) obtained by Algorithm 1, the first approximation Pareto optimal solutions of problem (1) can be achieved by the following algorithm.

**Algorithm 2.**

**Step 1.** Discrete induced set \( A^0 \), denote the discrete set by \( A^0_s \), and let \( s = 0 \).

**Step 2.** Divide the set \( A^0 \) into two subsets \( A^0_{s1} \) and \( A^0_{s2} \) randomly; that is \( A^0_s = A^0_{s1} \cup A^0_{s2} \).

**Step 3.** Input \( x_i \in A^0_{s1} \) and \( x_j \in A^0_{s2} \) into subnetwork \( A \) and subnetwork \( B \), respectively.

**Step 4.** Select Pareto optimal solutions.

**Step 4.1.** If \( O_2(S_q) = 1 \), then \( A^0_{s1} = A^0_{s1} \setminus x_j \). Let \( i = i + 1 \), go to Step 4.

**Step 4.2.** If \( O_2(S_q) = -1 \), then \( A^0_{s2} = A^0_{s2} \setminus x_j \). Let \( j = j + 1 \), go to Step 4.

**Step 4.3.** If \( O_2(S_q) = 0 \), then \( x_i, x_j \in A^0_{s+1} \). Go to Step 4.

**Step 5.** If \( A^0_{s1} = \phi \), \( A^0_{s2} \neq \phi \), then \((\forall x_j \in A^0_{s2}) \subset A^0_{s+1} \).

**Step 6.** If \( A^0_{s1} \neq \phi \), \( A^0_{s2} = \phi \), then \((\forall x_i \in A^0_{s1}) \subset A^0_{s+1} \).

**Step 7.** If the stopping criterion is met, then stop. Otherwise, go to Step 2.

In Step 2, the two subsets \( A^0_{s1} \) and \( A^0_{s2} \) have approximately equal number of feasible solutions. The output of Algorithm 2 is the first approximation Pareto optimal solution set of problem (1); we denote the set by \( P(A^0) \).

### 3.3. The Algorithm for BLBOP

Based on Algorithms 1 and 2, as well as the refinement strategy used by the literature [9] which is employed in this paper, the algorithm for solving the BLBOP can be described as follows.

**Algorithm 3.**

**Step 1.** Based on the induced set \( A^0 \) obtained by Algorithm 1, we determine the approximation Pareto optimal solution set \( P(A^0) \) of problem (1) by Algorithm 2.

**Step 2.** Let \( i = 0 \) and choose the distance \( y^i > 0 \).

**Step 3.** For any point \((x, y) \in P(A^0)\), determine the refinement induced set around this point by the refinement strategy according to (5.4) in [55].

**Step 4.** Set \( A^{i+1} = A^i \cup O_2(A^i) \); update the approximation Pareto optimal solution set \( A^{i+1} \) by Algorithm 2.

**Step 5.** If approximation of solution set of problem (1) is sufficient, then stop. Otherwise set \( i = i + 1 \), choose \( y^i > 0 \), and go to Step 3.

### 4. Results

In this section, we considered seven numerical examples and a practical problem to illustrate the feasibility of the proposed algorithm for problem (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

**4.1.1. Generational Distance (GD).** This metric used by Deb [65] is employed in this paper as a way of evaluating the closeness between the obtained Pareto optimal front and the theoretical Pareto optimal front. The 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 \( n \) is the number of the obtained Pareto optimal solutions by the proposed algorithm and \( d_i \) is the Euclidean distance between each obtained Pareto optimal solution and the nearest member of the theoretical Pareto optimal set.

**4.1.2. 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 [65]:

\[
SP = \frac{\sum_{m=1}^{M} d_m^n + \sum_{i=1}^{M} (\bar{d} - d_i)^2}{\sum_{m=1}^{M} d_m^n + n\bar{d}^2},
\]

where \( d_i = \min_{j}(|F_1(x, y) - F_1^j(x, y)| + |F_2(x, y) - F_2^j(x, y)|) \), \( i, j = 1, 2, \ldots, n \), \( \bar{d} \) is the mean of all \( d_i \), \( d_m^n \) 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, and \( n \) is the number of the obtained solutions by the proposed algorithm.

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

#### 4.2. Numerical Experiment

In this section, we will present seven BLBOPS to illustrate the proposed algorithm for the bilevel biobjective programming. Problem 1 and problem
2 are low-dimensional problems. Problems 3–6 are high-dimensional problems. For problem 7, the theoretical optimal front is unknown. In this paper, we refined every upper problem’s feasible solutions three times and the obtained results are compared with the classical literature.

Example 1. Example 1 is taken from [57]. Here \( x \in \mathbb{R}^1, y \in \mathbb{R}^2 \). In this example, the parameters \( y^0, y^1, y^2 \) of Algorithm 3 are 0.23, 0.11, and 0.03.

\[
\min_x F(x, y) = (y_1 - x, y_2) \\
\text{s.t. } G_i(y) = 1 + y_1 + y_2 \geq 0
\]

\[
\min_y f(x, y) = (y_1, y_2) \\
\text{s.t. } g_i(x, y) = x^2 - y_1^2 - y_2^2 \geq 0, \\
-1 \leq y_1, y_2 \leq 1, \quad 0 \leq x \leq 1.
\] (9)

Figure 1(a) shows the obtained Pareto front by the proposed algorithm and the method in [57]. Table 1 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 [57] with respect to spacing. By looking at the obtained Pareto fronts, some nondominated vectors produced by the method in [57] 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; that is, \( y_1 = -1 - y_2, y_2 = -1/2 \pm (1/\sqrt{8}) \sqrt{8x^2 - 4}, \) and \( x \in (1/\sqrt{2}, 1) \). However, some solutions obtained by the method in [57] do not meet the relationship.

Example 2. Example 2 is taken from [57]. Here \( x \in \mathbb{R}^1, y \in \mathbb{R}^2 \). In this example, the parameters \( y^0, y^1, y^2 \) of Algorithm 3 are 0.3, 0.21, and 0.12.

\[
\min_x F(x, y) = (x^2 + (y_1 - 1)^2 + y_2^2, (x - 1)^2 + (y_1 - 1)^2 + y_2^2) \\
\min_y f(x, y) = (y_1^2 + y_2^2, (y_1 - x)^2 + y_2^2) \\
-1 \leq x, y_1, y_2 \leq 2.
\] (10)

Figure 2(a) shows the obtained Pareto front by the proposed algorithm and the method in [57]. Obviously, both the proposed algorithm and the method in [57] almost have the same spacing. However, there are some areas of the Pareto optimal solution obtained by the method in [57] which is sparse. In Figures 2(a) and 2(b), it can be seen that the obtained solutions by the proposed algorithm almost follow the relationship; that is, \( x = y_1, y_1 \in [0.5, 1], y_2 = 0 \). However, some areas of the solutions obtained by the method in [57] are sparse and some solutions do not meet the relationship.

Example 3. Example 3 is taken from [60]. Here \( x \in \mathbb{R}^{10}, y \in \mathbb{R}^{10} \). In this example, the parameters \( y^0, y^1, y^2 \) of Algorithm 3 are 0.5, 0.36, and 0.12.

\[
\min_x F(x, y) = \left(1 + r - \cos(ax_i)\right) + \sum_{j=1}^{K} \left(x_j - \frac{j-1}{2}\right)^2 + \sum_{i=1}^{K} \left(y_i - x_i\right)^2 - y \cos \left(\frac{\pi x_i}{2y_1}\right). (1 + r - \sin(ax_i)) + \sum_{i=1}^{K} \left(x_j - \frac{j-1}{2}\right)^2 + r \sum_{i=1}^{K} \left(y_i - x_i\right)^2 - y \sin \left(\frac{\pi x_i}{2y_1}\right)\right),
\]

\[
\min_y f(x, y) = \left(y_1^2 + \sum_{i=1}^{K} (y_i - x_i)^2 + \sum_{i=1}^{10} \left(1 - \cos \left(\frac{\pi}{K} (y_i - x_i)\right)\right) + \sum_{i=1}^{K} (y_i - x_i)^2 + \sum_{i=1}^{10} \left(1 - \sin \left(\frac{\pi}{K} (y_i - x_i)\right)\right)\right)
\]

s.t. \( -K \leq y_i \leq K, \quad (i = 1, 2, \ldots, K); \)

\( 1 \leq x_i \leq 4, \)

\( -K \leq x_j \leq K, \quad (j = 2, 3, \ldots, K), \)

\( \alpha = 1, \)

\( r = 0.1, \)

\( r = 1, \)

\( y = 1, \)

\( K = 10.\) (11)

This problem is more difficult compared to the previous problems (Examples 1 and 2) because the lower level problem of this 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 [60] and the proposed algorithm in this paper. Tables 2 and 3 show the comparison of results between the two algorithms considering the metrics.
The obtained upper Pareto optimal front by the method in [57]

The obtained upper Pareto optimal front by the proposed algorithm

The theoretical Pareto optimal front

(a) The obtained Pareto front of Example 1

The obtained Pareto optimal solutions

The theoretical Pareto optimal solutions

(b) The obtained solution by the proposed algorithm

The obtained Pareto optimal solutions

The theoretical Pareto optimal solutions

(c) The obtained solution by [57]

**Figure 1**: The obtained Pareto front and solutions of Example 1.

| Prob. | GD The method in [57] | GD The proposed method | SP The method in [57] | SP The proposed method |
|-------|----------------------|------------------------|----------------------|------------------------|
| 1     | 0.00216               | 0.00097                | 0.0135               | 0.01201                |
| 2     | 0.01013               | 0.00312                | 0.00203              | 0.00969                |

**Table 1**: Results of the Generation Distance (GD) and Spacing (SP) metrics for Examples 1 and 2.

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 nondominated vectors produced by the method in [60] are not part of the true Pareto front and there are some areas are sparse. However, the proposed algorithm is able to cover the full Pareto optimal front. Furthermore, two obtained lower level Pareto optimal fronts are given when $x_1 = 2$ and $x_1 = 2.5$.

**Example 4.** Example 4 is taken from [60]. Here $x \in R^{10}$, $y \in R^{10}$. In this example, the parameters $\gamma^0, \gamma^1, \gamma^2$ of Algorithm 3 are 0.6, 0.38, and 0.16.

$$\min_{x} F(x, y)$$

$$= \nu_1(x_1) + \sum_{j=2}^{K} y_j^2 + 10 \left(1 - \cos \left(\frac{\pi}{K} y_j \right)\right)$$
The theoretical Pareto optimal front
○ The obtained Pareto optimal front by Algorithm 2
□ The obtained Pareto optimal front by the method in [57]

(a) The obtained Pareto front of Example 2

○ The obtained Pareto optimal solutions
□ The theoretical Pareto optimal solutions

(b) The obtained solutions by Algorithm 2

○ The obtained Pareto solutions
□ The theoretical Pareto solutions

(c) The obtained solutions by the method in [25]

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

Table 2: Results of the Generation Distance (GD) metrics for Examples 3, 4, 5, and 6.

| Prob. | The proposed algorithm | The method in [60] |
|-------|-----------------------|--------------------|
| 3     | 0.00019               | 0.00768            |
| 4     | 0.00015               | 0.06391            |
| 5     | 0.00038               | 0.01677            |
| 6     | 0.00021               | 0.00652            |

Table 3: Results of the Spacing (SP) metrics for Examples 3, 4, 5, and 6.

| Prob. | The proposed algorithm | The method in [60] |
|-------|-----------------------|--------------------|
| 3     | 0.00076               | 0.00197            |
| 4     | 0.00273               | 0.00269            |
| 5     | 0.00299               | 0.01737            |
| 6     | 0.00130               | 0.00127            |

\[
\min_y f(x, y) = \left( y_1^2 + \sum_{i=2}^{K} (y_i - x_i)^2 + \frac{1}{2} \sum_{i=1}^{K} y_i \right) \\
\text{s.t.} \quad -K \leq y_i \leq K, \quad (i = 1, 2, \ldots, K) \\
0.001 \leq x_i \leq 4, \\
-K \leq x_j \leq K, \quad (j = 2, 3, \ldots, K),
\]
\[ \alpha = 1, \quad r = 0.25, \quad \tau = 1, \quad \gamma = 4, \quad K = 10, \]
\[ (12) \]

where
\[
\begin{align*}
v_1(x_1) &= \begin{cases} 
\cos(0.2\pi) x_1 + \sin(0.2\pi) \sqrt{\left|0.02 \sin(5\pi x_1)\right|} & \text{for } 0 \leq x_1 \leq 1 \\
 x_1 - (1 - \sin(0.2\pi)) & \text{for } x_1 > 1
\end{cases} \\
v_2(x_1) &= \begin{cases} 
-\sin(0.2\pi) x_1 + \cos(0.2\pi) \sqrt{\left|0.02 \sin(5\pi x_1)\right|} & \text{for } 0 \leq x_1 \leq 1 \\
0.1 (x_1 - 1) - \sin(0.2\pi) & \text{for } x_1 > 1
\end{cases}
\end{align*} \]
\[ (13) \]

For Example 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 by the method in [60] and the proposed algorithm. Tables 2 and 3 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 [60] with respect to the generational distance though they almost have the same performance with respect to the spacing.

**Example 5.** Example 5 is taken from [60]. Here \( x \in R^{10}, \ y \in R^{10} \). In this example, the parameters \( y^0, y^1, y^2 \) of Algorithm 3 are 0.7, 0.49, and 0.23.

\[
\min_x F(x \cdot y)
\]
The obtained theoretical Pareto optimal front of lower level

The obtained Pareto optimal front by the proposed algorithm

The obtained Pareto optimal front by the method in [60]

Figure 4: The obtained Pareto front of Example 4.

\[
\begin{align*}
&= \left( x_1 + \sum_{j=3}^{K} \left( x_j - \frac{j}{2} \right)^2 \right) + r \sum_{i=3}^{K} (y_i - x_i)^2 \\
&- \cos \left( 4 \tan^{-1} \left( \frac{x_2 - y_2}{x_1 - y_1} \right) \right), \\
&x_2 + \sum_{j=3}^{K} \left( x_j - \frac{j}{2} \right)^2 + r \sum_{i=3}^{K} (y_i - x_i)^2 \\
&- \cos \left( 4 \tan^{-1} \left( \frac{x_2 - y_2}{x_1 - y_1} \right) \right)
\end{align*}
\]

s.t. \( G(x) = x_2 - \left( 1 - x_1^2 \right)^2 \geq 0 \)

\[
\begin{align*}
\min_{y} f(x, y) \\
&= \left( y_1 + \sum_{i=3}^{K} (y_i - x_i)^2, y_2 + \sum_{i=3}^{K} (y_i - x_i)^2 \right)
\end{align*}
\]

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

\( R(x_i) = \left( 0.1 + 0.15 \sin \left( 2\pi (x_1 - 0.1) \right) \right) \),

\(-K \leq y_i \leq K, \quad (i = 1, 2, \ldots, K), \)

\( 0 \leq x_i \leq K, \quad -K \leq x_j \leq K, \quad (i = 1, 2, \ldots, K), \)

\( \tau = 1, \)

\( r = 0.2, \)

\( K = 10. \)

Example 6. Example 6 is taken from [60]. Here \( x \in \mathbb{R}^1, \ y \in \mathbb{R}^9 \). In this example, the parameters \( \gamma_0, \gamma_1, \gamma_2 \) of Algorithm 3 are 0.5, 0.23, and 0.17.

\[
\begin{align*}
\min_{x} F(x, y) \\
&= \left( (1 - y_i) \left( 1 + \sum_{j=2}^{K} y_j^2 \right) x_1, y_1 \left( 1 + \sum_{j=2}^{K} y_j^2 \right) x_1 \right)
\end{align*}
\]

s.t. \( G_i(x, y) = - (1 - y_i) x_1 - \frac{1}{2} x_1 y_i \leq 1, \)

\[
\begin{align*}
\min_{y} f(x, y) \\
&= \left( (1 - y_i) \left( 1 + \sum_{j=K+1}^{K+L} y_j^2 \right) x_1, y_1 \left( 1 + \sum_{j=K+1}^{K+L} y_j^2 \right) x_1 \right)
\end{align*}
\]

s.t. \( 1 \leq x_1 \leq 2, \)

\(-1 \leq y_i \leq 1, \)

\(- (K + L) \leq y_j \leq (K + L) \)

\( (j = 2, 3, \ldots, K + L), \quad K = 5, \ L = 4. \)

Figure 5 shows the obtained Pareto front of Example 5 by the method in [60] and the proposed algorithm. Tables 2 and 3 show the comparison of results between the two algorithms considering the metrics previously described.

Figure 6 shows the obtained Pareto front of Example 6 by the method in [60] and the proposed algorithm. Tables 2 and 3
**Figure 6:** The obtained Pareto front of Example 6.

show the comparison of results between the two algorithms considering the metrics previously described. It can be seen that our algorithm and the method in [60] almost have the same performance of the spacing, but some nondominated vectors produced by the method in [60] are slightly off the true Pareto front. 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.

**Example 7.** Example 7 is taken from [55]. Here \( x \in \mathbb{R}^1 \), \( y \in \mathbb{R}^2 \). In this example, the parameters \( y^0 \), \( y^1 \), \( y^2 \) of Algorithm 3 are 0.3, 0.16, and 0.07, respectively.

![Graph](image)

**Figure 7** shows the final archive solutions by the proposed algorithm. For this problem, the exact Pareto optimal front is not known, but the obtained Pareto optimal front after four approximations by the proposed algorithm is similar to that reported in the previous study [55].

4.3. Application of the Algorithm for a Practical Problem. In a company, the CEO’s goal is usually to maximize net profits and quality of products, whereas a branch head’s goal is to maximize its own profit and worker satisfaction. The problem involves uncertainty and is bilevel in nature, as a CEO’s
The obtained Pareto optimal front by the proposed algorithm

Figure 7: The obtained Pareto optimal solutions with $A^3$.

The obtained Pareto optimal front by the proposed algorithm

The single optimal solution in [66]

Figure 8: The obtained Pareto front of the practical problem.

decision must take into account optimal decisions of branch heads. We present a deterministic version of the case study from [66] in (4). Figure 8 shows the obtained Pareto optimal front of this practical problem by the proposed algorithm. Note that Zhang et al. [66] only obtained a single optimal solution $x = (146.2955, 28.9394)$ and $y = (0, 67.9318, 0)$ which lies on the maximum of $F_2$ using weighted sum method. In contrast, a set of Pareto optimal solutions is obtained by the proposed algorithm. However, the fact that the single optimal solution in [66] is included in the obtained Pareto optimal solutions illustrates the feasibility of proposed algorithm. In this problem, the parameters $\gamma^0, \gamma^1, \gamma^2$ of Algorithm 3 are 0.2, 0.11, and 0.02.

\[
\begin{align*}
\max_x & \quad F(x, y) = (x_1 + 9x_2 + 10y_1 + y_2 + 3x_3, 9x_1 + 2x_2 + 2y_1 + 7y_2 + 4x_3) \\
\text{s.t.} & \quad g_1(x, y) = 3x_1 + 9x_2 + 9y_1 + 5y_2 + 3y_3 \leq 1039 \\
& \quad g_2(x, y) = -4x_1 - x_2 + 3y_1 - 3y_2 + 2y_3 \leq 94 \\
\min_y & \quad f(x, y) = (4x_1 + 6x_2 + 7y_1 + 4y_2 + 8y_3, 6x_1 + 4x_2 + 8y_1 + 7y_2 + 4y_3) \\
\text{s.t.} & \quad g_1(x, y) = 3x_1 - 9x_2 - 9y_1 - 4y_2 \leq 61 \\
& \quad g_2(x, y) = 5x_1 + 9x_2 + 10y_1 - y_2 - 2y_3 \leq 924 \\
& \quad g_3(x, y) = 3x_1 - 3x_2 + y_2 + 5y_3 \leq 420 \\
& \quad x_1, x_2, y_1, y_2, y_3 \geq 0.
\end{align*}
\]
5. Conclusions

In this paper, a two-stage ANN based on scalarization method is presented for solving BLBOPs. Seven numerical examples and a practical problem are used to state 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 easy to implement, which provides another appealing method for further study on the general BLMPP.

Conflicts of Interest

The authors declared that they have no conflicts of interest related to this work.

Acknowledgments

This work is supported by the National Science Foundation of China (61673006), the Young Project of Hubei Provincial Department of Education (Q20141304), and the Dr. Start-Up Fund by the Yangtze University (2014).

References

[1] L. S. Vicente and P. H. Calamai, “Bilevel and multilevel programming: a bibliography review,” Journal of Global Optimization, vol. 5, no. 3, pp. 291–306, 1994.
[2] S. Dempe, “Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints,” Optimization. A Journal of Mathematical Programming and Operations Research, vol. 52, no. 3, pp. 333–359, 2003.
[3] B. T. Colson, P. Marcotte, and G. Savard, “Bilevel programming: a survey,” 4OR: A Quarterly Journal of Operations Research, vol. 3, no. 2, pp. 87–107, 2005.
[4] B. T. Colson, P. Marcotte, and G. Savard, “An overview of bilevel optimization,” Annals of Operations Research, vol. 153, pp. 235–256, 2007.
[5] K. Shimizu, Y. Ishizuka, and J. F. Bard, Nondifferentiable and two-level mathematical programming, Kluwer Academic Publishers, Boston, Mass, USA, 1997.
[6] J. F. Bard, Practical Bilevel Optimization: Algorithms and Applications, vol. 30, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
[7] A. Migdalas, P. M. Pardalos, and P. Varbrand, Eds., Multilevel Optimization: Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
[8] S. Dempe, Foundations of Bilevel Programming, vol. 61, Kluwer Academic Publishers, London, UK, 2002.
[9] R. G. Jeroslow, “The polynomial hierarchy and a simple model for competitive analysis,” Mathematical Programming, vol. 32, no. 2, pp. 146–164, 1985.
[10] O. Ben-Ayed and C. E. Blair, “Computational difficulties of bilevel linear programming,” Operations Research, vol. 38, no. 3, pp. 556–560, 1990.
[11] J. F. Bard, “Some properties of the bilevel programming problem,” Journal of Optimization Theory and Applications, vol. 68, no. 2, pp. 371–378, 1991.
[12] L. Vicente, G. Savard, and J. Júdice, “Descent approaches for quadratic bilevel programming,” Journal of Optimization Theory and Applications, vol. 81, no. 2, pp. 379–399, 1994.
[13] J. F. Bard, “An algorithm for solving the general bilevel programming problem,” Mathematics of Operations Research, vol. 8, no. 2, pp. 260–272, 1983.
[14] T. A. Edmunds and J. F. Bard, “Algorithms for nonlinear bilevel mathematical programs,” Institute of Electrical and Electronics Engineers. Transactions on Systems, Man, and Cybernetics, vol. 21, no. 1, pp. 83–89, 1991.
[15] M. A. Amouzegar, “A global optimization method for nonlinear bilevel programming problems,” IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics, vol. 29, no. 6, pp. 771–777, 1999.
[16] J. B. Eto Etoa, “Solving quadratic convex bilevel programming problems using a smoothing method,” Applied Mathematics and Computation, vol. 217, no. 15, pp. 6680–6690, 2011.
[17] J. F. Bard and J. E. Falk, “An explicit solution to the multi-level programming problem,” Computers and Operations Research, vol. 9, no. 1, pp. 77–100, 1982.
[18] K. Shimizu and E. Aiyoshi, “A new computational method for Stackelberg and min-max problems by use of a penalty method,” Institute of Electrical and Electronics Engineers. Transactions on Automatic Control, vol. 26, no. 2, pp. 460–466, 1981.
[19] E. Aiyoshi and K. Shimizu, “A solution method for the static constrained Stackelberg problem via penalty method,” Institute of Electrical and Electronics Engineers. Transactions on Automatic Control, vol. 29, no. 12, pp. 1113–1114, 1984.
[20] Y. Ishizuka and E. Aiyoshi, “Double penalty method for bilevel optimization problems,” Annals of Operations Research, vol. 34, no. 1-4, pp. 73–88, 1992.
[21] Y. Iv, T. Hu, G. Wang, and Z. Wan, “A penalty function method based on Kuhn-Tucker condition for solving linear bilevel programming,” Applied Mathematics and Computation, vol. 188, no. 1, pp. 808–813, 2007.
[22] G. Savard and J. Gauvin, “The steepest descent direction for the nonlinear bilevel programming problem,” Operations Research Letters, vol. 15, no. 5, pp. 265–272, 1994.
[23] J. E. Falk and J. Liu, “On bilevel programming. Part I: general nonlinear cases,” Mathematical Programming, vol. 70, no. 1, Ser. A, pp. 47–72, 1995.
[24] R. Mathieu, L. Pittard, and G. Anandalingam, “Genetic algorithm based approach to bi-level linear programming,” Operations Research, vol. 28, no. 1, pp. 1–21, 1994.
[25] S. R. Hejazi, A. Memariani, G. Jahanshahloo, and M. M. Sepehri, “Linear bilevel programming solution by genetic algorithm,” Computers and Operations Research, vol. 29, no. 13, pp. 1913–1925, 2002.
[26] Y.-P. Wang, Y.-C. Jiao, and H. Li, “An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme,” IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews, vol. 35, no. 2, pp. 221–232, 2005.
[27] G.-m. Wang, X.-j. Wang, Z.-p. Wan, and S.-h. Jia, “An adaptive genetic algorithm for solving bilevel linear programming problem,” Applied Mathematics and Mechanics. English Edition, vol. 28, no. 12, pp. 1605–1612, 2007.
[28] H. I. Calvete, C. Galé, and P. M. Mateo, “A new approach for solving linear bilevel problems using genetic algorithms,” European Journal of Operational Research, vol. 188, no. 1, pp. 14–28, 2008.
[29] K. M. Lan, U. P. Wen, H.-S. Shih, and E. Lee, “A hybrid neural network approach to bilevel programming problems,” *Applied Mathematics Letters. An International Journal of Rapid Publication*, vol. 20, no. 8, pp. 880–884, 2007.

[30] Y. Lv, T. Hu, G. Wang, and Z. Wan, “A neural network approach for solving nonlinear bilevel programming problem,” *Computers & Mathematics with Applications. An International Journal*, vol. 55, no. 12, pp. 2823–2829, 2008.

[31] S. B. Yaakob and J. Watada, “Double-Layered Hybrid Neural Network Approach for Solving Mixed Integer Quadratic Bilevel Problems,” in *Integrated Uncertainty Management and Applications*, vol. 68 of *Advances in Intelligent and Soft Computing*, pp. 221–230, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[32] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity.” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[33] I. B. Pyne, “Linear programming on an electronic analog computer,” *Transactions of the American Institute Electrical Engineers*, vol. 75, pp. 139–143, 1956.

[34] Z. Sheng, Z. Lv, and R. Xu, “A new algorithm based on the Frank-Wolfe method and neural network for a class of bilevel decision making problems,” *Acta Automatic Sinica*, vol. 22, no. 6, pp. 657–665, 1996.

[35] H.-S. Shih, U.-P. Wen, E. S. Lee, K.-M. Lan, and H.-C. Hsiao, “A neural network approach to multiobjective and multilevel programming problems,” *Computers & Mathematics with Applications. An International Journal*, vol. 48, no. 1-2, pp. 95–108, 2004.

[36] Y. Lv, Z. Chen, and Z. Wan, “A neural network for solving a convex quadratic bilevel programming problem,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 2, pp. 505–511, 2010.

[37] X. Li, P. Tian, and X. Min, “A hierarchical particle swarm optimization for solving bilevel programming problems,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4029, pp. 1169–1178, 2006.

[38] R. J. Kuo and C. C. Huang, “Application of particle swarm optimization algorithm for solving bi-level linear programming problem,” *Computers & Mathematics with Applications. An International Journal*, vol. 58, no. 4, pp. 678–685, 2009.

[39] Y. Jiang, X. Li, C. Huang, and X. Wu, “Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bilevel programming problem,” *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4332–4339, 2013.

[40] Y. Gao, G. Zhang, J. Lu, and H.-M. Wee, “Particle swarm optimization for bi-level pricing problems in supply chains,” *Journal of Global Optimization*, vol. 51, no. 2, pp. 245–254, 2011.

[41] G. Zhang, G. Zhang, Y. Gao, and J. Lu, “Competitive strategic bidding optimization in electricity markets using bilevel programming and swarm technique,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2138–2146, 2011.

[42] S. B. Yaakob and J. Watada, “A hybrid intelligent algorithm for solving the bilevel programming models,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6277, no. 2, pp. 485–494, 2010.

[43] R. J. Kuo and Y. S. Han, “A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem—a case study on supply chain model,” *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 3905–3917, 2011.

[44] Z. Wan, G. Wang, and B. Sun, “A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems,” *Swarm and Evolutionary Computation*, vol. 8, pp. 26–32, 2013.

[45] 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*, vol. 88, no. 3, pp. 563–571, 1996.

[46] M. Gendreau, P. Marcotte, and G. Savard, “A hybrid tabu-ascent algorithm for the linear bilevel programming problem,” *Journal of Global Optimization*, vol. 8, no. 3, pp. 217–233, 1996.

[47] J. Rajesh, K. Gupta, H. S. Kusumakar, V. K. Jayaraman, and B. D. Kulkarni, “A Tabu Search Based Approach for Solving a Class of Bilevel Programming Problems in Chemical Engineering,” *Journal of Heuristics*, vol. 9, no. 4, pp. 307–319, 2003.

[48] K. H. Sahin and A. R. Ciric, “A dual temperature simulated annealing approach for solving bilevel programming problems,” *Computers and Chemical Engineering*, vol. 23, no. 1, pp. 11–25, 1999.

[49] H. I. Calvete, C. Galé, and M. Oliveros, “Bilevel model for production distribution planning solved by using ant colony optimization,” *Computers & Operations Research*, vol. 38, no. 1, pp. 320–327, 2011.

[50] Y. Gao, G. Zhang, J. Ma, and J. Lu, “A λ-cut and goal-programming-based algorithm for fuzzy-linear multiple-objective bilevel optimization,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 1–13, 2010.

[51] X. Shi and H. Xia, “Interactive bilevel multi-objective decision making,” *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 943–949, 1997.

[52] X. Shi and H. S. Xia, “Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers,” *Journal of Multi-Criteria Decision Analysis*, vol. 10, no. 1, pp. 27–34, 2001.

[53] M. A. Abo-Sinna and I. A. Baky, “Interactive balance space approach for solving multi-level multi-objective programming problems,” *Information Sciences*, vol. 177, no. 16, pp. 3397–3410, 2007.

[54] I. Nishizaki and M. Sakawa, “Stackelberg solutions to multiobjective bilevel decision making problems,” *Journal of Heuristics*, vol. 9, no. 4, pp. 307–319, 2003.

[55] G. Eichfelder, “Multiobjective bilevel optimization,” *Mathematical Programming. A Publication of the Mathematical Programming Society*, vol. 123, no. 2, Ser. A, pp. 419–449, 2010.

[56] K. Deb and A. Sinha, “Constructing test problems for bilevel evolutionary multi-objective optimization,” in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, CEC 2009, pp. 1153–1160, nior, May 2009.

[57] K. Deb and A. Sinha, “Solving bilevel multi-objective optimization problems using evolutionary algorithms,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5467, pp. 110–124, 2010.

[58] K. Deb and A. Sinha, “An evolutionary approach for bilevel multi-objective problems,” *Communications in Computer and Information Science*, vol. 35, pp. 17–24, 2009.

[59] A. Sinha and K. Deb, “Towards Understanding Evolutionary Bilevel Multi-Objective Optimization Algorithm,” *IFAC Proceedings Volumes*, vol. 42, no. 2, pp. 338–343, 2009.
[60] K. Deb and A. Sinha, “An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm,” *Evolutionary Computation*, vol. 18, no. 3, pp. 403–449, 2010.

[61] A. Sinha, “Bilevel multi-objective optimization problem solving using progressively interactive EMO,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6576, pp. 269–284, 2011.

[62] T. Zhang, T. Hu, Y. Zheng, and X. Guo, "An improved particle swarm optimization for solving bilevel multiobjective programming problem," *Journal of Applied Mathematics*, Article ID 626717, Art. ID 626717, 13 pages, 2012.

[63] T. Zhang, T. Hu, J.-W. Chen, Z. Wan, and X. Guo, "Solving bilevel multiobjective programming problem by elite quantum behaved particle swarm optimization," *Abstract and Applied Analysis*, vol. 2012, Article ID 102482, 20 pages, 2012.

[64] T. Zhang, T. Hu, X. Guo, Z. Chen, and Y. Zheng, "Solving high dimensional bilevel multiobjective programming problem using a hybrid particle swarm optimization algorithm with crossover operator," *Knowledge-Based Systems*, vol. 53, pp. 13–19, 2013.

[65] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[66] G. Zhang, J. Lu, and T. Dillon, "Decentralized multi-objective bilevel decision making with fuzzy demands," *Knowledge-Based Systems*, vol. 20, no. 5, pp. 495–507, 2007.