

Research Article

The Backpropagation Artificial Neural Network Based on Elite Particle Swarm Optimization Algorithm for Stochastic Linear Bilevel Programming Problem

Tao Zhang  and Xiaofei Li 

School of Information and Mathematics, Yangtze University, Jingzhou 434023, China

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

Received 24 July 2018; Accepted 8 October 2018; Published 22 October 2018

Academic Editor: Frederico R. B. Cruz

Copyright © 2018 Tao Zhang and Xiaofei Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For a class of stochastic linear bilevel programming problem, we firstly transform it into a deterministic linear bilevel covariance programming problem. Then, the deterministic bilevel covariance programming problem is solved by backpropagation artificial neural network based on elite particle swarm optimization algorithm (BPANN-PSO). Finally, we perform the simulation experiments and the results show that the computational efficiency of the proposed algorithm has a potential upside compared with the classical algorithm.

1. Introduction

The bilevel programming problem (BLP) is a nested optimization problem in which the feasible region of the upper level problem is determined implicitly by the solution set of the lower level problem. As an optimization tool, the BLP has been widely used in variety practical problems, for example, in homeland security [1–3], model production processes [4], the optimal tax policies formulation [5–7], the strategic for deregulating markets [8], and the optimization of retail channel structures [9]. In addition, the optimization theory of the BLP has been integrated in many other disciplines, such as in management [10, 11], chemical engineering [12, 13], structural optimization [14, 15], optimal control problems [16, 17], facility location [10, 18, 19], and transportation [20–22].

Therefore, many researchers are devoted to develop the algorithms for BLP and propose many efficient algorithms. Traditional methods commonly used to handle BLP include Karus-Kuhn-Tucker approach [23–26], Branch-and-bound method [27], and penalty function approach [28–31]. Despite a significant progress made in traditional optimization towards solving BLPPs, the properties such as differentiation and continuity are necessary for these algorithms.

Due to the limitation of the traditional algorithms, the heuristics such as evolutionary algorithms are recognized as potent tools for solving BLPPs. Mathieu et al. [32] firstly developed the genetic algorithm (GA) for bilevel linear programming problem. Motivated by the same reason, other kinds of GAs for BLPPs were also presented in [33–36]. Owing to its high speed of convergence and relative simplicity, the particle swarm optimization (PSO) algorithm has been employed for solving BLP problems recently [37–42].

However, it is worth noting that the papers mentioned above only focus on deterministic bilevel programming problem and the stochastic bilevel programming has seldom been studied so far. In 1999, Patriksson and Wynter [43] firstly proposed the stochastic mathematical programs with equilibrium constraints and introduced a framework for hierarchical decision-making problem under uncertainty. However, they did not give a numerical experiment. Gao, Liu, and Gen [44] presented a hybrid intelligent algorithm for a decentralized multilevel decision-making problem in stochastic environment in 2004. For the contracting arrangements of the long-term contracts and the spot markets transactions under uncertain electricity spot market, Wan et al. [45] proposed a stochastic bilevel programming model for the optimal bidding strategies between power seller and

buyer. They solved the model by Monte Carlo approximation method. It is worth noting that the decision variable in both levels are one-dimensional variable. Soon after, Wan, Fan, and Wang [46] proposed an interactive fuzzy decision-making method for the model in [45]. In 2013, He and Feng [47] presented an approximation algorithm for the compensated stochastic bilevel programming problem. In addition, the stability analysis and convergence analysis for bilevel stochastic programming problem can be seen in [48–50]. Obviously, they only researched the simple stochastic bilevel model and few of them have studied the numerical performance of the algorithm.

In this paper, we consider the general stochastic linear bilevel programming problem in which the coefficient of objective functions and the coefficient of constraint functions are random variables. For the problem, we firstly transformed it into a deterministic linear bilevel covariance programming problem with expected constraints, and then the deterministic bilevel covariance programming model is solved by the BPANN-PSO algorithm. Finally, we perform the simulation experiments and the results suggest that the variance obtained by our algorithm is better than the results in reference when the means of the upper objective function value is same. Furthermore, the computational efficiency of our algorithm performs better with the dimension increasing.

The rest of this paper is organized as follows. Section 2 introduces the definitions and properties of stochastic linear bilevel programming problem. Section 3 proposes the BPANN-PSO algorithm for stochastic linear bilevel programming problem. We use three test problems from the reference to measure and evaluate the proposed algorithm in Section 4, while the conclusion is reached in Section 5.

2. Stochastic Linear Bilevel Programming

Let $x \in R^{n_1}$, $y \in R^{n_2}$, $F : R^{n_1} \times R^{n_2} \rightarrow R$, $f : R^{n_1} \times R^{n_2} \rightarrow R$, $A \in m \times n_1$, $B \in m \times n_2$, we consider stochastic linear bilevel programming problem as follows:

$$\begin{aligned} \min_{x,y} \quad & F(x, y) = c_1^T x + d_2^T y \\ \min_y \quad & f(x, y) = c_2^T x + d_2^T y \\ \text{s.t.} \quad & Ax + By \leq b \\ & x \geq 0, y \geq 0. \end{aligned} \quad (1)$$

where $F(x, y)$ and $f(x, y)$ are the upper level and the lower level objective functions, respectively. The coefficient $c_1, c_2 \in R^{n_1}$, $d_1, d_2 \in R^{n_2}$ are the random variable.

Let α_i be a probability of the extent to which the i th constraint violation is admitted; $P(\cdot)$ means a probability measure. The i th constraint of problem (1) is interpreted as follows:

$$P(A_i x + B_i y \leq b_i) \geq \alpha_i \quad (i = 1, 2, \dots, m) \quad (2)$$

Inequality (2) means that the i th constraint may be violated, but at most $1 - \alpha_i$ proportion of the time. Let $F_i(\tau)$

be the distribution function of random variables b_i , then, inequality (2) is presented as

$$P(A_i x + B_i y \leq b_i) = 1 - F(A_i x + B_i y) \quad (3)$$

According to (2) and (3), we can obtain inequality (4),

$$F(A_i x + B_i y) \leq 1 - \alpha_i \quad (4)$$

Let $\tau = F^{-1}(1 - \alpha_i)$, $K_{1-\alpha_i} = \max \tau$. Then, from the monotonicity of the distribution function $F_i(\tau)$, inequality (2) is rewritten as

$$A_i x + B_i y \leq K_{1-\alpha_i} \quad (i = 1, 2, \dots, m) \quad (5)$$

That is,

$$Ax + By \leq K_{1-\alpha} \quad (6)$$

where $K_{1-\alpha} = (K_{1-\alpha_1}, K_{1-\alpha_2}, \dots, K_{1-\alpha_m})$. Based on (6), problem (1) is rewritten as the following problem with deterministic constraints:

$$\begin{aligned} \min_{x,y} \quad & F(x, y) = c_1^T x + d_1^T y \\ \min_y \quad & f(x, y) = c_2^T x + d_2^T y \\ \text{s.t.} \quad & Ax + By \leq K_{1-\alpha} \\ & x \geq 0, y \geq 0. \end{aligned} \quad (7)$$

To deal with the objective functions with random variables in both levels, the minimum covariance model [51] is applied in this paper. Then, problem (7) is rewritten as follows:

$$\begin{aligned} \min_{x,y} \quad & \text{var}(F(x, y)) = (x^T, y^T) V_1 \begin{pmatrix} x \\ y \end{pmatrix} \\ \min_y \quad & \text{var}(f(x, y)) = (x^T, y^T) V_2 \begin{pmatrix} x \\ y \end{pmatrix} \\ \text{s.t.} \quad & Ax + By \leq K_{1-\alpha} \\ & x \geq 0, y \geq 0. \end{aligned} \quad (8)$$

where $\text{var}(\cdot)$ denotes the variance of objective function; V_1 and V_2 are the covariance matrices of (c_1, d_1) and (c_2, d_2) , respectively. In this paper, we assume that V_1 and V_2 are positive-definite.

Though problem (8) can guarantee uniform distribution of the objective function values of both levels with a small drop, the decision makers in both levels often have their own expectation. Let β_1 and β_2 denote the means of the objective function values for the leader and the follower, respectively. Then, problem (8) is rewritten as the follows:

$$\begin{aligned} \min_{x,y} \quad & \text{var}(F(\bar{x}, y)) = (\bar{x}^T, y^T) V_1 \begin{pmatrix} \bar{x} \\ y \end{pmatrix} \\ \min_y \quad & \text{var}(f(\bar{x}, y)) = (\bar{x}^T, y^T) V_2 \begin{pmatrix} \bar{x} \\ y \end{pmatrix} \\ \text{s.t.} \quad & A\bar{x} + By \leq K_{1-\alpha} \end{aligned}$$

$$\begin{aligned}
m_1^c \bar{x} + m_1^d y &\leq \beta_1 \\
m_2^c \bar{x} + m_2^d y &\leq \beta_2 \\
x &\geq 0, \quad y \geq 0.
\end{aligned} \tag{9}$$

where m_1^c and m_1^d are the means of c_1 and d_1 , respectively. m_2^c and m_2^d are the means of c_2 and d_2 , respectively. For the fixed upper level decision variable \bar{x} , the rational reaction set $R(x)$ can be solved by problem (10).

$$\begin{aligned}
\min_y \quad & \text{var}(f(\bar{x}, y)) = (\bar{x}^T, y^T) V_2 \begin{pmatrix} \bar{x} \\ y \end{pmatrix} \\
\text{s.t.} \quad & A\bar{x} + By \leq K_{1-\alpha} \\
& m_1^c \bar{x} + m_1^d y \leq \beta_1 \\
& m_2^c \bar{x} + m_2^d y \leq \beta_2 \\
& y \geq 0.
\end{aligned} \tag{10}$$

Let $A'_1 = (A, m_1^c, m_1^d)^T$, $A'_2 = (B, m_2^c, m_2^d)^T$, $b' = (K_{1-\alpha}, \beta_1, \beta_2)^T$. Then, problem (9) is formulated as

$$\begin{aligned}
\min_{x,y} \quad & \text{var}(F(\bar{x}, y)) = (\bar{x}^T, y^T) V_1 \begin{pmatrix} \bar{x} \\ y \end{pmatrix} \\
\min_y \quad & \text{var}(f(\bar{x}, y)) = (\bar{x}^T, y^T) V_2 \begin{pmatrix} \bar{x} \\ y \end{pmatrix} \\
\text{s.t.} \quad & A'_1 \bar{x} + A'_2 y \leq b'
\end{aligned} \tag{11}$$

For problem (11), the basic notions of BLP are recalled as follows:

(a) Constraint region of the BLP:

$$S = \{(x, y) \mid A'_1 x + A'_2 y \leq b\} \tag{12}$$

(b) Feasible set for the lower level problem for each fixed x :

$$S(x) = \{(x, y) \mid A'_1 x + A'_2 y \leq b\} \tag{13}$$

(c) Projection of onto the upper level maker's decision space:

$$S(X) = \{x \mid \exists y, A'_1 x + A'_2 y \leq b\} \tag{14}$$

(d) The lower level maker's rational reaction set for each fixed $x \in S(X)$:

$$P(x) = \{y : y \in \arg \min [f(x, y) : y \in S(x)]\} \tag{15}$$

(e) Inducible region:

$$IR = \{(x, y) : (x, y) \in S, y \in P(x)\} \tag{16}$$

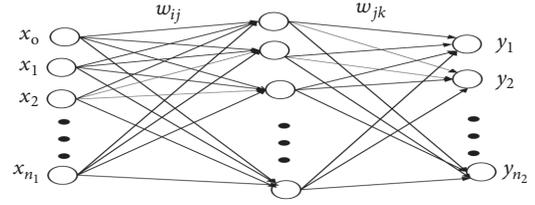


FIGURE 1: The diagram of BPANN.

Definition 1. A point (x, y) is feasible if $(x, y) \in IR$.

Definition 2. A feasible point (x^*, y^*) is an optimal solution if $(x^*, y^*) \in IR$ and $F(x^*, y^*) \leq F(\bar{x}, \bar{y})$, $\forall (\bar{x}, \bar{y}) \in IR$.

Definition 3. If (x^o, y^o) is the optimistic solution for problem (11), (x^o, y^o) is given by $\min_{x,y} \{F(x, y) \mid y \in P(x), A'_1 x + A'_2 y \leq b\}$.

3. The Algorithm

For problem (11), 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. Based on this fact, the BPANN-PSO algorithm may have a great potential for solving problem (11). In the following, the BPANN-PSO is presented for solving problem (11).

3.1. The Structure of BPANN. In this paper, the BPANN includes three layers: the input layer, the hidden layer, and the output layer. The number of nodes per layer is n_1 , m , and n_2 ($m = \sqrt{n_1 + n_2}$). The connectivity weight from the i -th node of the input to the j -th node of the hidden layer is w_{ij} . w_{jk} denotes the connectivity weight from the j -th node of the hidden layer to the k -th node of the output layer. Let the Sigmoid function $f(x) = 1/(1 + e^{-x})$. Figure 1 shows the diagram of the BPANN.

Let x_{ni} represent the i -th input value of the n -th sample. The output of the hidden layer is given by

$$s_{nj} = f\left(\sum_{i=0}^{n_1} w_{ij} x_{ni}\right), \quad j = 1, 2, \dots, m. \tag{17}$$

For the output layer, the output is given by

$$y_{nl} = f\left(\sum_{j=0}^m w_{jk} s_{nj}\right), \quad l = 1, 2, \dots, n_2. \tag{18}$$

$y_{nk}^{optimal}$ denotes the k -th node's expected output of the output layer. The squared differences between the computed

and desired outputs are given by (19) and the error sum of all N training pairs is given by (20):

$$E_n = \frac{1}{2} \sum_{l=1}^{n_2} (t_{nl} - y_{nl}) \quad (19)$$

$$E = \sum_{n=1}^N E_n = \frac{1}{2} \sum_{n=1}^N \sum_{l=1}^{n_2} (t_{nl} - y_{nl})^2 \quad (20)$$

3.2. The BPANN-PSO Algorithm. In this algorithm, the position of the particle represents the connectivity weight of the BPANN and the particles in the elite set A_t are the batch training samples. Based on this idea, the BPANN-PSO algorithm can be designed as follows.

Suppose $W_{ij} = \{w_{ij}\} = \{w_{01}, w_{11}, w_{21}, \dots, w_{n_1 1}; w_{02}, w_{12}, w_{22}, \dots, w_{n_1 2}; \dots; w_{0m}, w_{1m}, w_{2m}, \dots, w_{n_1 m}\}$, $W_{jk} = \{w_{jk}\} = \{w_{11}, w_{21}, \dots, w_{m1}; w_{12}, w_{22}, \dots, w_{m2}; \dots; w_{1n_2}, w_{2n_2}, \dots, w_{mn_2}\}$. The l -th particle in A_t can be presented as $A^l = \{X^l, Y^{l, optimal}\} = \{x_1^l, x_2^l, \dots, x_{n_1}^l; y_1^{l, optimal}, y_2^{l, optimal}, \dots, y_{n_2}^{l, optimal}\}$ ($l = 1, 2, \dots, N(A_t)$).

Algorithm 4.

Step 1. Initialize the population P with N particles and each particle can be presented as $P_p = \{W_{ij}^p; W_{jk}^p\}$ ($p = 1, 2, \dots, N$). Let the $N(A_t)$ be the number of the particles in the elite set A_t .

Step 2. Train the BPANN using the PSO.

Step 2.1. Calculate its fitness value according to (20).

Step 2.2. Determine its personal best particle and global best particle.

Step 2.3. Update particle's position and velocity.

Step 3 (stopping criterion). If $gbest(p^t) \leq \varepsilon = 1 \times 10^{-4}$, stop. Otherwise, go to step 2.

In step 2.3, the inertia weight $\omega = 0.5 + r/2.0$, $r = \text{rand}(0, 1)$, the acceleration constants $c_1 = c_2 = 1.49618$, and $r_1, r_2 \in \text{random}(0, 1)$.

3.3. The Algorithm for BLP. We proposed the BPANN-PSO for problem (11) and we update the upper level decision variables using the method in [52]. However, the way to solve the lower level problems is completely different. In reference [52], for each updated upper level decision variable, we need to reexecute the lower level optimization problem by cooperative coevolution PSO (CCPSO). In other words, the historical optimal solutions do not contribute to the current optimal solution. In our algorithm, for the updated upper level decision variable, we can predict the corresponding lower level optimal solution directly by BPANN. From the above analysis, we can see that the computational efficiency of our proposed algorithm performs well than the method in the reference [52]. Furthermore, we also give the basic working framework for these two algorithms (see Figure 2)

Algorithm 5.

Step 1. Initialization scheme. Initialize a random population (N_u) of the upper level variable X .

Step 2. For the fixed X^i ($i = 1, 2, \dots, N_u$), initialize a random population (N_l) of the lower level variable Y . Then, we perform a lower level optimization procedure to determine the corresponding lower level optimal solution $Y_i^{optimal}$ using PSO.

Step 3. Evaluate the fitness value of the complete solutions $Z_i = (X_i, Y_i^{optimal})$ ($i = 1, 2, \dots, N_u$) by the upper level function F and constraints conditions. Copy all the members in an elite set A_t .

Step 4. Update the upper level decision variables using the simulated binary crossover operator (SBX) and the polynomial variation method (PM).

Step 5. For the new upper level variable \bar{X}^i . Predict the lower level optimal solutions using Algorithm 4.

Step 6. Update the elite set A_t . Copy all the offspring from the previous step to the elite set A_t and choose the best $|A_t|$ members as the new members of A_t .

Step 7. Perform a termination check. If the termination check is false, go to step 4.

In the Algorithm 5, the PSO parameters are set as follows: the inertia weight $\omega = 0.7298$, acceleration coefficients $c_1 = c_2 = 1.49618$, and $r_1, r_2 \in \text{random}(0, 1)$, $N = 20$. At step 7, the termination criteria based on variance is described as follows. For each upper level variable i , the variance at the j -th generation is v_j^i . Then, the stop metrics α is computed as

$$\alpha_j = \sum_{i=1}^{N_u} \frac{v_j^i}{v_0^i} \quad (21)$$

In this paper, the value of α_{stop} is set as 10^{-4} for the upper level.

3.4. Algorithm Complexity Analysis. By the algorithm in [51], for the vector u_i with dimension $(m+n_2+2)$, each component of the vector has two states, zero or nonzero, therefore, the vector has a total of 2^{m+n_2+2} forms. We examine the case where the calculation is the most expensive here, that is, until the last combination satisfies the complementary relaxation condition, suppose the number of vertices is K and let the time be t_1 for computing the $u_i g_i(x^k, y^k)$, and the total time required to complete a check is as follows: $T_1 = K \times 2^{m+n_2+2} \times t_1$.

In Algorithm 4 of our paper, the number of nodes of the input layer, the hidden layer, and the output layer are n_1 , m , and n_2 ($m = \sqrt{n_1 + n_2}$), respectively. The batch training sample is $N(A_t)$ and the time for computing multiplication is t_2 ; for one training sample, the computing time is as follows: $T_2 = (n_1 \times m + m \times n_2) \times t_2 = (n_1 \times \sqrt{n_1 + n_2} + n_2 \times \sqrt{n_1 + n_2}) \times t_2 = (n_1 + n_2)^{3/2} \times t_2$. The total computing time of all samples is as follows: $T_3 = N(A_t) \times (n_1 + n_2)^{3/2} \times t_2$. In Algorithm 5,

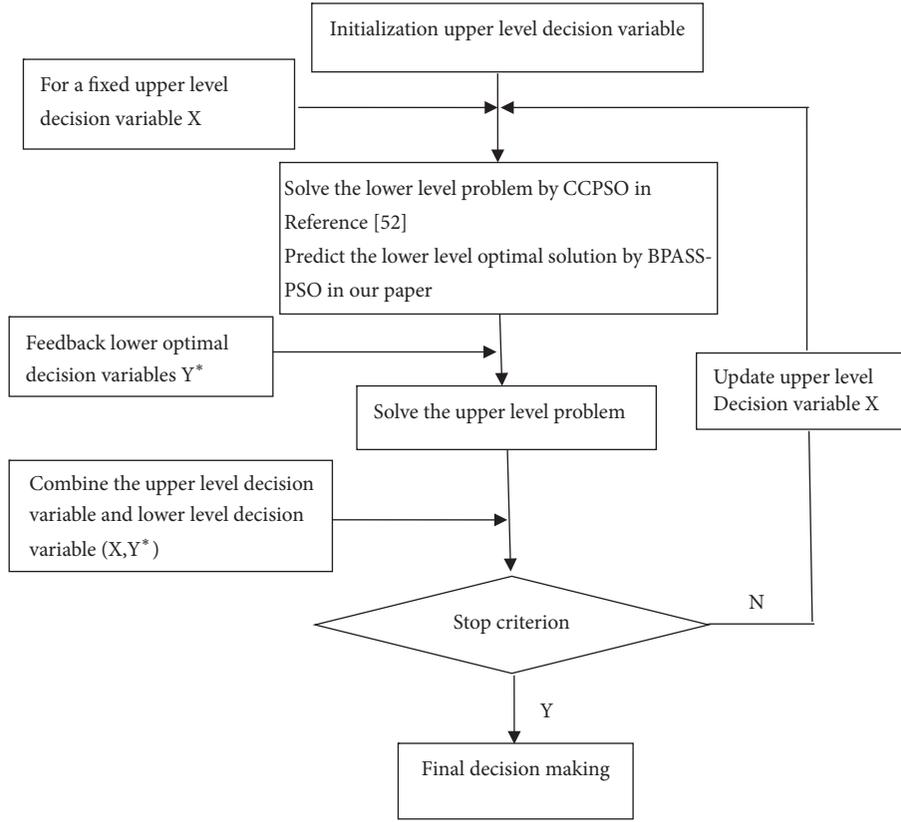


FIGURE 2: The basic working framework for two algorithms.

 TABLE 1: The mean, variance, and acceptable probability of b .

	b_1	b_2	b_3	b_4	b_5
mean	50.11	113.15	15.16	13.16	25.63
variance	9.0	36	9.0	4.0	16
acceptable probability	0.85	0.7	0.9	0.7	0.8

suppose the number of iterations is G , the computing time required to solve problem (11) is as follows: $T_4 = G \times N(A_t) \times (n_1 + n_2)^{3/2} \times t_2$. t_1 and t_2 are almost the same when they have the same operating platform. Comparing T_1 and T_4 , our algorithm has obvious advantages when the dimension of the decision variable increases.

4. Numerical Experiment

In this section, we test the algorithm using three examples. All results presented in this paper have been obtained on a personal computer (CPU: AMD Phenom(tm)IIX6 1055T 2.80 GHz; RAM: 3.25 GB) using a C# implementation of the proposed algorithm.

Example 1. Let $x \in R^1$, $y \in R^1$; we consider the problem from [51]

$$\begin{aligned} \min_{x,y} F(x, y) &= c_1 x + d_1 y \\ \min_y f(x, y) &= c_2 x + d_2 y \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & -x + 3y \leq b_1, \\ & 10x - y \leq b_2 \\ & 3x + y \geq b_3, \\ & x + 2y \geq b_4 \\ & 3x + 2y \geq b_5, \\ & x \geq 0, y \geq 0. \end{aligned} \tag{22}$$

where $m_1^c = -2.0$, $m_1^d = -3.0$, $m_2^c = 2.0$, $m_2^d = 1.0$. $V_1 = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$, $V_2 = \begin{pmatrix} 1 & -1 \\ -1 & 6 \end{pmatrix}$. All the right-hand side values $b = (b_1, b_2, \dots, b_5)$ are normal random variables. The mean, variance, and probability of b are shown in Table 1.

According to the above conditions, problem (22) can be rewritten as follows:

$$\min_{x,y} F(x, y) = (x, y) \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

TABLE 2: The comparison results of Example 1 from 11 runs.

Algorithm	means constrain	optimal solutions	means		variance		CPU time (s)
			UL	LL	UL	LL	
BPANN-PSO	Y	(5.15, 6.90)	-31.00	17.20	266.95	241.11	0.0335
Method in [51]	Y	(5.17, 6.89)	-31.00	17.23	267.12	240.32	0.0879
BPANN-PSO	N	(6.99, 4.00)	-25.98	17.99	201.64	88.94	0.0173
Method in [51]	N	(7.00, 4.00)	-26.00	18.00	202.00	89.00	0.0208

$$\begin{aligned}
 \min_y \quad & f(x, y) = (x, y) \begin{pmatrix} 1 & -1 \\ -1 & 6 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
 \text{s.t.} \quad & -x + 3y \leq 47, \\
 & 10x - y \leq 110 \\
 & 3x + y \geq -19, \\
 & x + 2y \geq -15 \\
 & 3x + 2y \geq -29, \\
 & -2x - 3y \leq -31 \\
 & 2x + y \leq 33, \\
 & x \geq 0, y \geq 0.
 \end{aligned} \tag{23}$$

We solve problem (23) with constrain of means of the objective function values and we also solved problem (23) without this constrain. From Table 2, we can see that the variance obtained by our algorithm is better than the results in [51] when the means of the upper objective function value is the same. Moreover, the computational efficiency of our algorithm performs better.

Example 2. Let $x \in R^5, y \in R^3, c_1, c_2 \in R^{5 \times 1}, d_1, d_2 \in R^{3 \times 1}, A \in R^{3 \times 5}, B \in R^{3 \times 3}, b \in R^{3 \times 1}$; we consider problem from [51]

$$\begin{aligned}
 \min_{x,y} \quad & F(x, y) = c_1^T x + d_1^T y \\
 \min_y \quad & f(x, y) = c_2^T x + d_2^T y \\
 \text{s.t.} \quad & Ax + By \leq b, \\
 & x \geq 0, y \geq 0
 \end{aligned} \tag{24}$$

where $b_1 \sim N(220, 4^2), b_2 \sim N(145, 3^2), b_3 \sim N(-18, 5^2), \alpha_1 = 0.85, \alpha_2 = 0.95, \alpha_3 = 0.90, K_{1-\alpha_1} = 215.85, K_{1-\alpha_2} = 140.07, K_{1-\alpha_3} = 24.41, \beta_1 = 60.0, \text{ and } \beta_2 = 35.0.$ The expectation of random variable c_1, c_2, d_1, d_2 is shown in Table 3. The coefficient matrices of constraints are as follows:

$$A = \begin{pmatrix} 7 & 2 & 6 & 9 & 11 \\ 5 & 6 & -4 & 3 & 3 \\ -4 & -7 & -2 & -6 & -8 \end{pmatrix},$$

TABLE 3: The expectation of random variables c_1, c_2, d_1, d_2 .

m_{c_1}	5	2	1	2	1	m_{d_1}	6	1	3
m_{c_2}	10	-7	1	-2	-5	m_{d_2}	3	-4	6

$$B = \begin{pmatrix} 4 & 3 & 8 \\ -7 & -1 & -3 \\ -3 & -5 & -6 \end{pmatrix}.$$

V_1

$$= \begin{pmatrix} 16.0 & -1.6 & 1.8 & -3.5 & 1.3 & -2.0 & 4.0 & -1.4 \\ -1.6 & 25.0 & -2.2 & 1.6 & -0.7 & 0.5 & -1.3 & 2.0 \\ 1.8 & -2.2 & 25.0 & -2.0 & 5.0 & -2.4 & 1.2 & -2.1 \\ -3.5 & 1.6 & -2.0 & 16.0 & -2.0 & 3.0 & 2.2 & 2.8 \\ 1.3 & -0.7 & 5.0 & -2.0 & 4.0 & -1.0 & 0.8 & -2.0 \\ -2.0 & 0.5 & -2.4 & 3.0 & -1.0 & 1.0 & -1.5 & 0.6 \\ 4.0 & -1.3 & 1.2 & 2.2 & 0.8 & -1.5 & 4.0 & -2.3 \\ -1.4 & 2.0 & -2.1 & 2.8 & -2.0 & 0.6 & -2.3 & 4.0 \end{pmatrix}$$

V_2

$$= \begin{pmatrix} 4.0 & -1.4 & 0.8 & 0.2 & 1.6 & 1.0 & 1.2 & 2.0 \\ -1.4 & 4.0 & 0.2 & -1.0 & -2.2 & 0.8 & 0.9 & 1.8 \\ 0.8 & 0.2 & 9.0 & 0.2 & -1.5 & 1.5 & 1.0 & 0.6 \\ 0.2 & -1.0 & 0.2 & 36.0 & 0.8 & 0.4 & -1.5 & 0.7 \\ 1.6 & -2.2 & -1.5 & 0.8 & 25.0 & 1.2 & -0.2 & 2.0 \\ 1.0 & 0.8 & 1.5 & 0.4 & 1.2 & 25.0 & 0.5 & 1.4 \\ 1.2 & 0.9 & 1.0 & -1.5 & -0.2 & 0.5 & 9.0 & 0.8 \\ 2.0 & 1.8 & 0.6 & 0.7 & 2.0 & 1.4 & 0.8 & 16.0 \end{pmatrix}$$

(25)

According to model (8), problem (24) can be rewritten as follows:

$$\min_{x,y} \quad F(x, y) = (x^T, y^T) V_1 \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\min_y \quad f(x, y) = (x, y) V_2 \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\text{s.t.} \quad Ax + By \leq (215.85, 140.07, 24.41)^T$$

TABLE 4: The comparison results of Example 2 from 11 runs.

Algorithm	means constrain	optimal solutions	means		variance		CPU time (s)
			UL	LL	UL	LL	
BPANN-PSO	Y	(0.0529, 0.2509, 0, 0.0072, 0.9501; 0.5987, 1.2321, 1.1399)	9.9769	-2.2822	5.9689	77.7253	4.3898
Method in [51]	Y	(0.0530, 0.2510, 0, 0.0070, 0.9500; 0.5990, 1.2320, 1.1400)	9.9770	-2.2820	5.9700	77.7250	21.2172
BPANN-PSO	N	(0.4101, 0.6320, 0.0671, 0.2702, 0.8012; 0.2757, 0.9731, 0.7488)	9.5979	-3.3727	20.6536	48.3552	0.9654
Method in [51]	N	(0.4100, 0.6320, 0.0670, 0.2700, 0.8010, 0.2760, 0.9730, 0.7490)	9.5980	-3.3720,	20.6540	48.3550	7.1836

TABLE 5: comparison results of Example 3 from 11 runs.

	algorithm	optimal solutions	the optimal value		CPU run time (s)
			UL	LL	
10 sample model	BPANN-PSO	(64.8839, 18.9248)	-102.73335	78.4070	0.00179
	Method in [47]	(64.8840, 18.9250)	-102.7340	78.4070	0.0379
30 sample model	BPANN-PSO	(75.2431, 18.8569)	-112.9569	56.9433	0.00201
	Method in [47]	(75.2432, 18.8570)	-112.9572	56.9395	0.0463

$$\begin{aligned}
 m_{c_1}^T x + m_{d_1}^T y &\leq 60.0 \\
 m_{c_2}^T x + m_{d_2}^T y &\leq 35.0
 \end{aligned}
 \tag{26}$$

For problem (26), we solve the problem with constrain of the means of the objective function values and we also solved it without this constrain. From Table 4, we can see that the variance obtained by our algorithm is better than the results in [51] when the mean of the upper objective function value is almost the same. Furthermore, the computational efficiency of our algorithm performs better with the number of dimension increasing.

Examples 1 and 2 are the general linear stochastic bilevel programming problem; that is to say, the coefficient of objective functions and the coefficient of constraint functions are random variables. Furthermore, the algorithm is also effective for the stochastic linear bilevel programming with chance constants. We consider the compensated stochastic linear bilevel programming as follows.

Example 3. Let $x \in R^1, y \in R^1$, we consider the problem from [47]

$$\begin{aligned}
 \min_{x,y} \quad & F(x, y) = -x - 2y \\
 \min_y \quad & f(x, y) = -x + 11y \\
 \text{s.t.} \quad & P \left(\begin{array}{l} A_1x - B_1y \leq 4 \\ 2A_2x - B_2y \leq 24 \\ 3A_3 + 4B_3y \leq 96 \\ A_4x + 7B_4y \leq 126 \end{array} \right) \geq 0.8, \\
 & x \geq 0, y \geq 0.
 \end{aligned}
 \tag{27}$$

where $A_1, A_2, \dots, A_4, B_1, B_2, \dots, B_4 \sim U(0, 1), p = 0.8$. In this paper, we set the 10 samples model and 30 samples model for problem (27), then, we obtained problem (28) and (29):

10 samples for model:

$$\begin{aligned}
 \min_{x,y} \quad & F(x, y) = -x - 2y \\
 \min_y \quad & f(x, y) = -x + 11y \\
 \text{s.t.} \quad & 0.516x - 1.6y \leq 3.2 \\
 & 1.133x - 0.5441y \leq 19.2 \\
 & 1.608x + 2.139y \leq 76.8 \\
 & 0.528x + 3.516y \leq 100.8 \\
 & x \geq 0, y \geq 0.
 \end{aligned}
 \tag{28}$$

30 samples for model:

$$\begin{aligned}
 \min_{x,y} \quad & F(x, y) = -x - 2y \\
 \min_y \quad & f(x, y) = -x + 11y \\
 \text{s.t.} \quad & 0.4325x - 1.556y \leq 3.2 \\
 & 0.793x - 0.4260y \leq 19.2 \\
 & 1.2186x + 1.6556y \leq 76.8 \\
 & 0.4865x + 3.4041y \leq 100.8 \\
 & x \geq 0, y \geq 0.
 \end{aligned}
 \tag{29}$$

We solve problem (28) and problem (29) by our algorithm and the method in [47]. From Table 5, we can see that the

computational efficiency of our algorithm performs better when they have almost the same optimal solutions.

5. Conclusion and Future Works

In this paper, we designed the BPANN-PSO algorithm to solve the general stochastic linear bilevel programming problem and three test problems from the reference are used to measure and evaluate the proposed algorithm. The results suggest that the variance obtained by our algorithm is better than the results in reference when the mean of the upper objective function value is almost the same. Furthermore, the computational efficiency of our algorithm performs better with the number of the dimensions increasing.

In the future works, we will further discuss how to efficiently use the infeasible solution with good performance near the optimal. This kind of discussion could improve the performance of our BPANN-PSO, particularly when the optimal front lies on the boundaries between the feasible and infeasible regions.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61673006) and China Scholarship Council (20170842011).

References

- [1] G. Brown, M. Carlyle, D. Diehl, J. Kline, and K. Wood, "A two-sided optimization for theater ballistic missile defense," *Operations Research*, vol. 53, no. 5, pp. 745–763, 2005.
- [2] L. M. Wein, "Homeland security: From mathematical models to policy implementation: The 2008 Philip McCord Morse lecture," *Operations Research*, vol. 57, no. 4, pp. 801–811, 2009.
- [3] B. An, F. Ordóñez, M. Tambe et al., "A deployed quantal response-based patrol planning system for the u.s. coast guard," *Interfaces*, vol. 43, no. 5, pp. 400–420, 2013.
- [4] M. G. Nicholls, "Aluminum production modeling—a nonlinear bilevel programming approach," *Operations Research*, vol. 43, no. 2, pp. 208–218, 1995.
- [5] M. Labbé, P. Marcotte, and G. Savard, "A bilevel model of taxation and its application to optimal highway pricing," *Management Science*, vol. 44, no. 12, pp. 1608–1622, 1998.
- [6] A. Sinha, P. Malo, A. Frantsev, and K. Deb, "Multi-objective Stackelberg game between a regulating authority and a mining company: A case study in environmental economics," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC '13*, pp. 478–485, Mexico, June 2013.
- [7] A. Sinha, P. Malo, and K. Deb, "Transportation policy formulation as a multi-objective bilevel optimization problem," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC '15*, pp. 1651–1658, Japan, May 2015.
- [8] X. Hu and D. Ralph, "Using EPECs to model bilevel games in restructured electricity markets with locational prices," *Operations Research*, vol. 55, no. 5, pp. 809–827, 2007.
- [9] N. Williams, P. K. Kannan, and S. Azarm, "Retail channel structure impact on strategic engineering product design," *Management Science*, vol. 57, no. 5, pp. 897–914, 2011.
- [10] H. Sun, Z. Gao, and J. Wu, "A bi-level programming model and solution algorithm for the location of logistics distribution centers," *Applied Mathematical Modelling*, vol. 32, no. 4, pp. 610–616, 2008.
- [11] J. Bard, "Coordination of multi-divisional firm through two levels of management," *Omega*, vol. 11, no. 5, pp. 457–465, 1983.
- [12] W. Smith and R. Missen, *Chemical Reaction Equilibrium Analysis: Theory and Algorithms*, John Wiley & Sons, New York, NY, USA, 1982.
- [13] P. A. Clark and A. W. Westerberg, "Bilevel programming for steady-state chemical process design-I. Fundamentals and algorithms," *Computers & Chemical Engineering*, vol. 14, no. 1, pp. 87–97, 1990.
- [14] M. P. Bendsoe, "Optimization of structural topology, shape, and material," Technical report, 1995.
- [15] S. Christiansen, M. Patriksson, and L. Wynter, "Stochastic bilevel programming in structural optimization," *Structural and Multidisciplinary Optimization*, vol. 21, no. 5, pp. 361–371, 2001.
- [16] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous Robots*, vol. 28, no. 3, pp. 369–383, 2010.
- [17] S. Albrecht, K. Ramírez-Amaro, F. Ruiz-Ugalde et al., "Imitating human reaching motions using physically inspired optimization principles," in *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS '11*, pp. 602–607, Slovenia, October 2011.
- [18] Q. Jin and S. Feng, "Bi-level simulated annealing algorithm for facility location," *Systems Engineering*, vol. 2, 2007.
- [19] T. Uno, H. Katagiri, and K. Kato, "An evolutionary multi-agent based search method for Stackelberg solutions of bilevel facility location problems," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 5, pp. 1033–1042, 2008.
- [20] A. Migdalas, "Bilevel programming in traffic planning: models, methods and challenge," *Journal of Global Optimization*, vol. 7, no. 4, pp. 381–405, 1995.
- [21] I. Constantin and M. Florian, "Optimizing frequencies in a transit network: a nonlinear bi-level programming approach," *International Transactions in Operational Research*, vol. 2, no. 2, pp. 149–164, 1995.
- [22] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard, "A bilevel model for toll optimization on a multicommodity transportation network," *Transportation Science*, vol. 35, no. 4, pp. 345–358, 2001.
- [23] J. F. Bard, "An algorithm for solving the general bilevel programming problem," *Mathematics of Operations Research*, vol. 8, no. 2, pp. 260–272, 1983.
- [24] T. A. Edmunds and J. F. Bard, "Algorithms for nonlinear bilevel mathematical programs," *The Institute of Electrical and Electronics Engineers Systems, Man, and Cybernetics Society*, vol. 21, no. 1, pp. 83–89, 1991.

- [25] M. A. Amouzegar, "A global optimization method for nonlinear bilevel programming problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 6, pp. 771–777, 1999.
- [26] J. B. Etoa Etoa, "Solving quadratic convex bilevel programming problems using a smoothing method," *Applied Mathematics and Computation*, vol. 217, no. 15, pp. 6680–6690, 2011.
- [27] J. F. Bard and J. E. Falk, "An explicit solution to the multi-level programming problem," *Computers & Operations Research*, vol. 9, no. 1, pp. 77–100, 1982.
- [28] 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.
- [29] E. Aiyoshi and K. Shimizu, "A solution method for the static constrained Stackelberg problem via penalty method," *IEEE Transactions on Automatic Control*, vol. 29, no. 12, pp. 1111–1114, 1984.
- [30] 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.
- [31] Y. Lv, 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.
- [32] 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.
- [33] S. R. Hejazi, A. Memariani, G. Jahanshahloo, and M. M. Sepehri, "Linear bilevel programming solution by genetic algorithm," *Computers & Operations Research*, vol. 29, no. 13, pp. 1913–1925, 2002.
- [34] 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, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 221–232, 2005.
- [35] 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.
- [36] 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.
- [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): Preface*, 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] 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): Preface*, vol. 6277, no. 2, pp. 485–494, 2010.
- [41] 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.
- [42] 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.
- [43] M. Patriksson and L. Wynter, "Stochastic mathematical programs with equilibrium constraints," *Operations Research Letters*, vol. 25, no. 4, pp. 159–167, 1999.
- [44] J. Gao, B. Liu, and M. Gen, "A Hybrid Intelligent Algorithm for Stochastic Multilevel Programming," *IEEJ Transactions on Electronics, Information and Systems*, vol. 124, no. 10, pp. 1991–1998, 2004.
- [45] Z. Wan, C. Xiao, X. Wang, K. Xiao, Y. Huang, and X. Peng, "Bilevel programming model of optimal bidding strategies under the uncertain electricity markets," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 28, no. 19, pp. 12–16, 2004.
- [46] Z. P. Wan, H. Fan, G. M. Wang, and X. Y. Peng, "An interactive fuzzy bidding strategy based on stochastic bilevel programming under uncertain electricity markets," *Engineering Journal of Wuhan University*, vol. 39, no. 5, pp. 85–90, 2006 (Chinese).
- [47] Y. He and C. Q. Feng, "An approximating algorithm for compensated bilevel stochastic programming," *Basic Sciences Journal of Textile Universities*, vol. 26, no. 1, pp. 110–113, 2013.
- [48] W. N. Zhou and Y. L. Huo, "Analysis of the Stability for the Solution Set to Stochastic Programming Approximation of Bilevel," *Journal of Chongqing Technology Business University*, vol. 30, no. 7, pp. 19–23, 2013.
- [49] W. N. Zhou, Y. L. Huo, and Z. Y. Hu, "The Hausdorff Convergence of the Optimal Solution Set of Approximation for Bilevel Stochastic Programming," *Journal of Natural Science of Hunan Normal University. Human Shifan Daxue Ziran Kexue Xuebao*, vol. 39, no. 3, pp. 80–83, 2016.
- [50] Y. Liu, H. Wang, Y. S. Xu, and Y. Li, "Convergence of approximate solutions in bi-level stochastic programming," *Pure and Applied Mathematics. Chuncui Shuxue yu Yingyong Shuxue*, vol. 24, no. 4, pp. 768–773, 2008.
- [51] S. Masatoshi and N. Ichiro, *Cooperative and Non-cooperative Multi-Level Programming*, Springer Science, New York, NY, USA, 2009.
- [52] T. Zhang, Z. Chen, and J. Chen, "A cooperative coevolution PSO technique for complex bilevel programming problems and application to watershed water trading decision making problems," *Journal of Nonlinear Sciences and Applications. JNSA*, vol. 10, no. 4, pp. 2115–2132, 2017.

