Hindawi
Complexity
Volume 2019, Article ID 1480392, 12 pages
https://doi.org/10.1155/2019/1480392

Research Article

The Improved Antlion Optimizer and Artificial Neural Network for Chinese Influenza Prediction

Hongping Hu,[1] **Yangyang Li**,[1] **Yanping Bai**,[1] **Juping Zhang**,[2] **and Maoxing Liu**[1]

[1]*School of Science, North University of China, Taiyuan, Shanxi 030051, China*
[2]*Complex Systems Research Center, Shanxi University, Taiyuan, Shanxi 030006, China*

Correspondence should be addressed to Hongping Hu; hhp92@163.com

Received 22 February 2019; Revised 20 June 2019; Accepted 11 July 2019; Published 5 August 2019

Academic Editor: Michele Scarpiniti

The antlion optimizer (ALO) is a new swarm-based metaheuristic algorithm for optimization, which mimics the hunting mechanism of antlions in nature. Aiming at the shortcoming that ALO has unbalanced exploration and development capability for some complex optimization problems, inspired by the particle swarm optimization (PSO), the updated position of antlions in elitism operator of ALO is improved, and thus the improved ALO (IALO) is obtained. The proposed IALO is compared against sine cosine algorithm (SCA), PSO, Moth-flame optimization algorithm (MFO), multi-verse optimizer (MVO), and ALO by performing on 23 classic benchmark functions. The experimental results show that the proposed IALO outperforms SCA, PSO, MFO, MVO, and ALO according to the average values and the convergence speeds. And the proposed IALO is tested to optimize the parameters of BP neural network for predicting the Chinese influenza and the predicted model is built, written as IALO-BPNN, which is against the models: BPNN, SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN. It is shown that the predicted model IALO-BPNN has smaller errors than other six predicted models, which illustrates that the IALO has potentiality to optimize the weights and basis of BP neural network for predicting the Chinese influenza effectively. Therefore, the proposed IALO is an effective and efficient algorithm suitable for optimization problems.

1. Introduction

Optimization problems exist in scientific research and engineering areas [1–3], such as statistical physics [4, 5], computer science [6], artificial intelligence [7], and pattern recognition [8].

For every optimization problem, there is at least one global optimal solution and there may be some local optimal solutions as well as a global optimal solution. Many researchers wish to seek the global optimum for solving optimization problems. Therefore, many methods are created and applied to solve optimization problems. In particular, swarm intelligence algorithms proposed give strong support. Genetic algorithm (GA) proposed by Holland in 1992 [9] simulates Darwinian evolution, and particle swarm optimization (PSO) proposed in 1995 [10] simulates birds' behavior. And GA algorithm and PSO algorithm are constantly improved and applied to many aspects, such as complex system [11], hyperlastic materials [12], radiation detectors [13], and reaction kinetic parameters of biomass pyrolysis [14].

Since then, swarm intelligence algorithms have been constantly proposed and widely applied to find the global optimum of the optimization problems in various fields. For example, Moth-flame optimization algorithm (MFO) [15] mimics the moth eventually converging towards the light. Multi-verse optimizer (MVO) [16] was proposed on the base of three concepts in cosmology: white hole, black hole, and wormhole. Sine cosine algorithm (SCA) [17] creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution using a mathematical model based on sine and cosine functions. Whale optimization algorithm (WOA) [18] mimics the social behavior of humpback whales. And its improvement proposed for global optimization consists of three strategies: the chaotic initialization phase, Gaussian mutation, and a chaotic local search with a "shrinking" strategy [19]. The antlion optimizer (ALO) [20] mimics the hunting mechanism of antlions in nature, which has been improved and applied into automatic generation control [21], cluster analysis [22], photovoltaic cell [23], power systems [24], and parameter estimation of

photovoltaic models [25]. Harris Hawks Optimizer (HHO) [26] proposed in 2019 is a novel population-based, nature-inspired optimization paradigm, whose main inspiration is the cooperative behavior and chasing style of Harris hawks in nature called surprise pounce. HHO is used to perform the function optimizations and the real-world engineering problems.

Swarm intelligence algorithms can be also applied into feature selection (FS), such as gravitational search algorithm (GSA) inspired by Newton's law of gravity which is combined with evolutionary crossover and mutation operators [27], an efficient optimizer based on the simultaneous use of the Grasshopper Optimization Algorithm (GOA), selection operators, and Evolutionary Population Dynamics (EPD) [28], the Binary Dragonfly Algorithm (BDA) using time-varying transfer functions [29], and binary Salp Swarm Algorithm (SSA) with asynchronous updating rules and a new leadership structure [30].

Swarm intelligence algorithms have been also applied to optimize the weights and basis of artificial neural networks for prediction and classification. For example, SCA and GA are used to optimize the weight and basis of artificial neural network for predicting the direction of stock market index, respectively [31, 32]. An improved dynamic particle swarm optimization with AdaBoost algorithm is used to optimize the parameters of generalized radial basis function neural network for stock market prediction [33], and an Improved Exponential Decreasing Inertia Weight-Particle Swarm Optimization Algorithm is utilized to optimize the parameters of radial basis function neural network for the air quality index (AQI) prediction [34], respectively. Artificial tree (AT) algorithm was improved and applied to optimize the parameters of artificial neural network for predicting influenza-like illness [35]. MVO algorithm was combined with PSO algorithm to optimize the parameters of Elman neural network for classification of endometrial carcinoma with gene expression [36]. Based on Gaussian mutation and a chaotic local search that are employed to increase the population diversity of MFO and the flame updating process of MFO for better exploiting the locality of the solutions, respectively, the proposed CLSGMFO approach [37] is used to perform the function optimizations and is combined with a hybrid kernel extreme learning machine (KELM) model for financial prediction. Based on opposition-based learning (OBL) and the drawbacks of GWO, OBLGWO [38] is proposed to tune the parameters of KELM for dealing with two real-world problems: second major selection (SMS) problem and thyroid cancer diagnosis (TCD) problem.

In this paper, the updated position of antlions inspired by PSO in elitism operator of ALO is improved and then the improved ALO (IALO) is obtained. The proposed IALO is compared against SCA, PSO, MFO, MVO, and ALO by performing on 23 classical benchmark functions. The results are that IALO is superior to SCA, PSO, MFO, MVO, and ALO according to the average values and the convergence speeds. Then, IALO is tested to optimize the parameters of BP neural network for predicting the Chinese influenza and the predicted model is built, written as IALO-BPNN, which is compared with the models: BPNN, SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN. The results illustrate that the IALO has potentiality to optimize the weights and basis of BP neural network for predicting the Chinese influenza effectively. Therefore, the proposed IALO is an effective and efficient algorithm for optimization.

The remaining of the paper is organized as follows. The original ALO is displayed in Section 2. The proposed IALO is described in Section 3. Section 4 shows the comparison results of SCA, PSO, MFO, MVO, ALO, and IALO algorithms for 23 benchmark functions, which show better search performances and fast convergence speeds. In Section 4, IALO is also utilized to optimize the parameters of BP neural network (BPNN) for predicting the Chinese influenza. Discussion is presented in Section 5. And conclusion and future direction are given in Section 6.

2. The Antlion Optimizer

Antlion optimizer (ALO) proposed by Seyedali Mirjalili in 2015 is a novel nature-inspired algorithm that mimics the hunting mechanism of antlions in nature. There are five main steps of hunting prey in ALO: the random walk of ants, building traps, entrapment of ants in traps, catching preys, and rebuilding taps. In fact, the ALO algorithm mimics interaction between antlions and ants in the trap, where ants are allowed to move stochastically for food in the search space and antlions hunt the ants using the traps. The matrices

$$M_{Ant} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1d} \\ A_{21} & A_{22} & \cdots & A_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n1} & A_{n2} & \cdots & A_{nd} \end{bmatrix}, \tag{1}$$

and

$$M_{Antlion} = \begin{bmatrix} AL_{11} & AL_{12} & \cdots & AL_{1d} \\ AL_{21} & AL_{22} & \cdots & AL_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ AL_{n1} & AL_{n2} & \cdots & AL_{nd} \end{bmatrix}, \tag{2}$$

denote the matrices for saving the position of n ants and n antlions, respectively.

Let f be the fitness (objective) function during optimization. The matrices

$$M_{OA} = \begin{bmatrix} f([A_{11}, A_{12}, \ldots, A_{1d}]) \\ f([A_{21}, A_{22}, \ldots, A_{2d}]) \\ \cdots \\ f([A_{n1}, A_{n2}, \ldots, A_{nd}]) \end{bmatrix}, \tag{3}$$

and

$$M_{OAL} = \begin{bmatrix} f([AL_{11}, AL_{12}, \ldots, AL_{1d}]) \\ f([AL_{21}, AL_{22}, \ldots, AL_{2d}]) \\ \cdots \\ f([AL_{n1}, AL_{n2}, \ldots, AL_{nd}]) \end{bmatrix}, \tag{4}$$

are the matrices for saving the fitness values of n ants and n antlions.

In ALO algorithm, there exist six operators as follows:

(i) Random Walks of Ants. Ants update their positions with random walk at every step of optimization. A random walk is based on the following:

$$X(t) = [0, cumsum(2r(t_1) - 1), \\ cumsum(2r(t_2) - 1), \ldots, cumsum(2r(t_n) - 1)], \quad (5)$$

where $r(t)$ is a stochastic function defined as follows:

$$r(t) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand \leq 0.5, \end{cases} \quad (6)$$

cumsum calculates the cumulative sum, n is the maximum number of iterations, t shows the current iteration, and *rand* is a random number generated with uniform distribution in the interval of [0,1]. Since every search space has a boundary (range of variable) in order to keep the random walks inside the search space, the ants are normalized using the following equation (min–max normalization) before updating position of ants:

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i, \quad (7)$$

where a_i is the minimum of random walk of i-th variable, d_i is the maximum of random walk in i-th variable, c_i^t is the minimum of i-th variable at t-th iteration, and d_i^t indicates the maximum of i-th variable at t-th iteration.

(ii) Trapping in Antlion's Pits. Random walks of ants are affected by antlions traps. Equations

$$c_i^t = Antlion_j^t + c^t, \quad (8)$$

and

$$d_i^t = Antlion_j^t + d^t, \quad (9)$$

show that ants randomly walk in a hypersphere defined by the vectors c and d around a selected antlion, where c^t is the minimum of all variables at t-th iteration, d^t indicates the vector including the maximum of all variables at t-th iteration, c_j^t is the minimum of all variables for i-th ant, d_j^t is the maximum of all variables for i-th ant, and $Antliont_j$ shows the position of the selected j-th antlion at t-th iteration.

(iii) Building Trap. A roulette wheel in ALO algorithm is employed to select the fitter antlions for catching ants based on the fitness value during optimization.

(iv) Sliding Ants towards Antlion. According to the fitness values, antlions can build traps proportionally and ants move randomly. Once the ants are in the trap, antlions shoot sands outwards the center of the pit. This behaviour slides down the trapped ant that is trying to escape, where the radius of ants random walks hypersphere is decreased adaptively. The equations

$$c^t = \frac{c^t}{I}, \quad (10)$$

and

$$d^t = \frac{d^t}{I}, \quad (11)$$

shrink the radius of updating ants positions and mimic sliding process of ant inside the pits, where I is a ratio, c^t is the minimum of all variables at t-th iteration, and d^t indicates the vector including the maximum of all variables at t-th iteration.

In (10) and (11), $I = 10^w(t/T)$, where T is the maximum number of iterations and w is a constant defined based on the current iteration t ($w = 2$ when $t > 0.1T$, $w = 3$ when $t > 0.5T$, $w = 4$ when $t > 0.75T$, $w = 5$ when $t > 0.9T$, and $w = 6$ when $t > 0.95T$, shown in Figure 1). Basically, the constant w can adjust the accuracy level of exploitation.

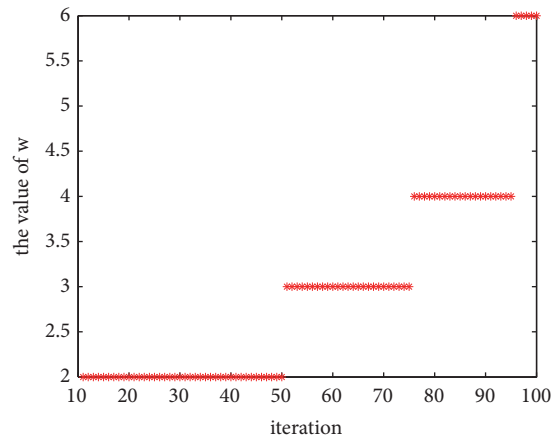

Figure 1: The value of the constant w.

(v) Catching Prey and Rebuilding the Pit. When an ant reaches the bottom of the pit, it is caught by an antlion. The antlion updates its position to the latest position of the hunted ant to enhance its chance of catching new prey as follows:

$$Antlion_j^t = Ant_i^t \quad if\ f(Ant_i^t) > f(Antlion_j^t) \quad (12)$$

where $Antlion_j^t$ shows the position of selected j-th antlion at t-th iteration and Ant_i^t indicates the position of i-th ant at t-th iteration.

(vi) Elitism. Elitism is an important characteristic of evolutionary algorithms in order to maintain the best solution(s) obtained at any stage of optimization process. In ALO algorithm, the best antlion is considered to be an elite. Every ant randomly walks around a selected antlion by the roulette wheel and the elite simultaneously as follows:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (13)$$

where R_A^t is the random walk around the antlion selected by the roulette wheel at t-th iteration, R_E^t is the random walk around the elite at t-th iteration, and Ant_i^t indicates the position of i-th ant at t-th iteration.

Let A be a function that generates the random initial solutions, B manipulates the initial population provided by the function A, and C returns true when the end criterion is satisfied. Based on the above proposed operators, the ALO algorithm is defined as a three-tuple as follows:

$$ALO\,(A, B, C) \tag{14}$$

where the functions A, B, and C are defined as follows:

$$\phi \longrightarrow^{A} \{M_{Ant}, M_{OA}, M_{Antlion}, M_{OAL}\} \tag{15}$$

$$\{M_{Ant}, M_{Antlion}\} \longrightarrow^{B} \{M_{Ant}, M_{Antlion}\} \tag{16}$$

$$\{M_{Ant}, M_{Antlion}\} \longrightarrow^{C} \{true, false\} \tag{17}$$

where M_{Ant} is the matrix of the position of ants, $M_{Antlion}$ includes the position of antlions, M_{OA} contains the corresponding fitness of ants, and M_{OAL} has the fitness of antlions.

3. The Improved ALO Algorithm

In PSO, there are two updated methods: the velocity update and the position update of the particle. The updated velocity and position of every particle in PSO are shown as follows:

$$\begin{aligned} v_i^{t+1} &= \omega v_i^t + c_1 rand\,() \left(pbest^t - x_i^t\right) \\ &\quad + c_2 rand\,() \left(gbest^t - x_i^t\right), \end{aligned} \tag{18}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{19}$$

where v_i^t and x_i^t are the current velocity and position of the i-th particle at the t-th iteration, c_1 and c_2 are acceleration coefficients that control the influence of the $pbest^t$ and $gbest$ on the search process, respectively, $rand()$ is a random number in [0,1], $pbest^t$ is the current best position of all the particles at the t-th iteration, $gbest$ is the best position among all the particles at whole iterations, and ω is the inertia weight which is the nonnegative constant less than 1.

The structure of (18) in PSO is employed as the updated approach of the elitism operator of ALO, where the adopted method is that x_i^t, $pbest^t$, $gbest$, v_i^t, and v_i^{t+1} are replaced by $elite$, R_A^t, R_E^t, $(R_A^t + R_E^t)/2$ in (13), and Ant_i^t, respectively. Thus the improved elitism operator of ALO is obtained as follows:

$$\begin{aligned} Ant_i^t &= \omega \frac{R_A^t + R_E^t}{2} + c_1 rand\,() \left(R_A^t - elite\right) \\ &\quad + c_2 rand\,() \left(R_E^t - elite\right). \end{aligned} \tag{20}$$

Thus the ALO is improved, named by IALO.

The concrete steps of the IALO algorithm are as follows.

Step 1. Initialize the population of ants and antlions randomly. Calculate the fitness of ants and antlions. Find the best antlions and assume it as the elite (determined optimum).

Step 2. For every ant, select an antlion using Roulette wheel, update c and d using (10) and (11), create a random walk and normalize it using (5) and (7), and update the position of ant using (20).

Step 3. Calculate the fitness of all ants. Replace an antlion with its corresponding ant if it becomes fitter (12). Update elite if an antlion becomes fitter than the elite.

Step 4. If the end criterion is satisfied, return the elite. Otherwise, return to Step 2.

4. Experiments

4.1. Experiments on 23 Classic Benchmark Functions

4.1.1. Benchmark Functions. In this section, 23 classic benchmark functions are selected to perform the IALO algorithm. Table 1 not only shows 9 unimodal functions $F_1(x) - F_9(x)$, 7 multimodal functions $F_{10}(x) - F_{16}(x)$, and 7 fixed-dimension benchmark functions $F_{17}(x) - F_{23}(x)$, but also shows the concrete expressions, the dimension, and the minimum function value of these 23 benchmark functions, where the varied dimensions of 16 benchmark functions $F_1(x) - F_{16}(x)$ are all taken 30 and the minimum values of these classic benchmark functions are all 0 except functions $F_{10}(x)$ and $F_{16}(x)$, and the dimensions of functions $F_{17}(x) - F_{23}(x)$ are fixed.

As their names imply, every unimodal function has single optimum and every multimodal function has more than one optimum. One of the optima is called global optimum and the rest are called local optima. Therefore, solving an optimized problem is to seek the global optimum and to avoid the local optimum. From Table 1, the global optima are 0 except the functions $F_{10}(x), F_{16}(x) - F_{23}(x)$, while the minimum value of $F_{10}(x)$ is $-418.9829 \times Dim$, which is changed based on the number of variables (Dim), the minimum value of $F_{16}(x)$ is -1, the dimension of functions $F_{17}(x) - F_{23}(x)$ is fixed, and the minimum values of functions $F_{17}(x) - F_{23}(x)$ are $1, 0.00030, -1.0316, 0.398, 3, -3.86$, and -3.32, respectively. Figure 2 shows typical 2D plots of six benchmark functions $F_1(x)$, $F_2(x)$, $F_{10}(x)$, $F_{13}(x)$, $F_{17}(x)$, and $F_{21}(x)$.

For verification of the validness of the proposed algorithm, IALO, SCA, PSO, MFO, MVO, and ALO are employed for comparison with IALO. All algorithms in this section are conducted under the same conditions. Each algorithm is run 30 times independently on these 23 classic benchmark functions and the maximum iteration number of every run is 1000, and the average value and the standard deviation of the best approximated solution in the last iteration are taken as the criteria.

In (20), we take the parameters:

$$\omega = e^{-t/T}, \tag{21}$$

$$c_1 = c_2 = 2, \tag{22}$$

where t is the current iteration and T is the maximum iteration.

Table 1: Description of benchmark function.

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[-100,100]	0
$F_4(x) = \max_i \{ \lvert x_i \rvert, \ 1 \le i \le n \}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} \left([x_i + 0.5] \right)^2$	30	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[-1.28,1.28]	0
$F_8(x) = \left(x_1 - 1 \right)^2 + \sum_{i=2}^{n} i \left(2x_i^2 - x_{i-1} \right)^2$	30	[-10,10]	0
$F_9(x) = \sum_{i=1}^{n} i x_i^2$	30	[-10,10]	0
$F_{10}(x) = \sum_{i=1}^{n} (-x_i \sin \left(\sqrt{\lvert x_i \rvert} \right)$	30	[-500,500]	$-418.9829 \times Dim$
$F_{11}(x) = \sum_{i=1}^{n} \left[x_i^2 - 10 \cos \left(2\pi x_i \right) + 10 \right]$	30	[-5.12,5.12]	0
$F_{12}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^{n} \cos \left(2\pi x_i \right) \right) + 20 + \exp$	30	[-32,32]	0
$F_{13}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[-600,600]	0
$F_{14}(x) = \frac{\pi}{n} \left\{ 10 \sin \left(\pi y_1 \right) + \sum_{i=1}^{n-1} \left(y_i - 1 \right)^2 \left[1 + 10 \sin^2 \left(\pi y_{i+1} \right) \right] + \left(y_n^2 - 1 \right)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[-50,50]	0
$y_i = 1 + \frac{x_i + 1}{4}$			
$u(x_i, a, k, m) = \begin{cases} k \left(x_i - a \right)^m & x_i > a \\ 0 & -a < x_i < a \\ k \left(-x_i - a \right)^m & x_i < -a \end{cases}$			
$F_{15}(x) = 0.1\{\sin^2 \left(3\pi x_1 \right) + \sum_{i=1}^{n} \left(x_i - 1 \right)^2 \left[1 + \sin^2 \left(3\pi x_i + 1 \right) \right] + (x_n - 1)^2 \left[1 + \sin^2 (2\pi x_n \right] \} + \sum_{i=1}^{n} u \left(x_i, 5, 100, 4 \right)$	30	[-50,50]	0
$F_{16}(x) = \left\{ \left[\sum_{i=1}^{n} \sin^2 \left(x_i \right) \right] - \exp \left(- \sum_{i=1}^{n} x_i^2 \right) \right\} \cdot \exp \left[- \sum_{i=1}^{n} \sin^2 \sqrt{\lvert x_i \rvert} \right]$	30	[-10,10]	-1
$F_{17}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(x_i - a_{ij} \right)^6} \right)^{-1}$	2	[-65,65]	1
$F_{18}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 \left(b_i^2 + b_i x_2 \right)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$F_{19}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{20}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[-5,5]	0.398
$F_{21}(x) = \left[1 + \left(x_1 + x_2 + 1 \right)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2 \right) \right] \times \left[30 + \left(2x_1 - 3x_2 \right)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2 \right) \right]$	2	[-2,2]	3
$F_{22}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij} \right)^2$	3	[1,3]	-3.86
$F_{23}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij} \left(x_j - p_{ij} \right)^2$	6	[0,1]	-3.32

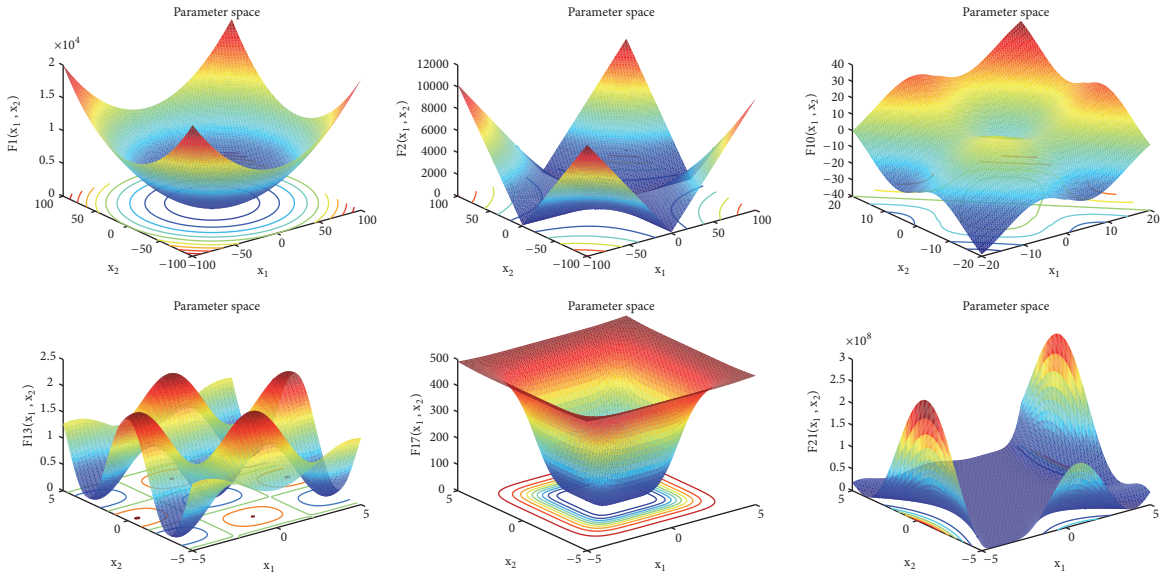

FIGURE 2: 2D version of typical benchmark functions $F_1, F_2, F_{10}, F_{13}, F_{17}, F_{21}$.

And in PSO algorithm, the inertia weight ω is taken as (21).

4.1.2. Results on Benchmark Functions. Based on the above settings, we perform the algorithms: SCA, PSO, MFO, MVO, ALO, and IALO for comparison on the 23 classic benchmark functions $F_1(x) - F_{23}(x)$ shown in Table 1.

We calculate the average value (Avg.) and the standard derivation (Std.) of the final iterations of 30 times' performances. The comparative results are obtained by SCA, PSO, MFO, MVO, ALO, and IALO on the 23 classic benchmark functions, shown in Table 2.

From Table 2, it can be seen that the obtained average values of all unimodal functions $F_1(x) - F_9(x)$ except function $F_6(x)$ performed by IALO algorithm for 30 times are the least among these six algorithms and they are $5.6713E - 09, 2.1869E - 06, 8.1060E - 08, 1.9699E - 05, 6.0874E + 00, 2.4135E - 04, 3.5681E - 01, 6.5564E - 10$, respectively. That is, IALO obtains the best solutions on the functions $F_1(x) - F_5(x), F_7(x) - F_9(x)$. Hence, it is concluded that IALO is better than the other comparative algorithms: SCA, PSO, MFO, MVO, and ALO in dealing with the 9 unimodal functions.

Table 2 shows that the obtained average values of all multimodal functions $F_{10}(x) - F_{16}(x)$ except function $F_{14}(x) - F_{15}(x)$ performed by IALO algorithm for 30 times are the least among these six algorithms and they are $-1.2465E + 04, 4.2863E - 09, 2.1607E - 05, 1.0333E - 08$, and $-9.9996E - 01$, respectively. That is, IALO obtains the best solutions on the functions $F_{10}(x) - F_{13}(x), F_{16}(x)$. Hence, it is shown that IALO is better than the other comparative algorithms: SCA, PSO, MFO, MVO, and ALO in dealing with the 7 multimodal functions.

In dealing with the 7 fixed-dimension multimodal functions $F_{17}(x) - F_{23}(x)$ in Table 1, IALO obtains the best solution on function F_{18}, shown in Table 2. We observe from Table 2 that PSO has the minimum average values on $F_8(x), F_{14}(x), F_{15}(x), F_{19}(x) - f_{21}(x)$ and they are $2.7838E - 07, 2.6382E - 09, 4.0291E - 03, -1.0316E + 00, 39789E - 01, 3.0000E + 00$. And we also observe from Table 2 that MFO has the minimum average values on $F_{19}(x), F_{20}(x), F_{22}(x), F_{23}(x)$ and they are $-1.0316E + 00, 39789E - 01, -3.8628E + 00, -3.2220E + 00$. And MVO has the minimum average value $9.9800E - 01$ on $F_{17}(x)$.

Therefore, IALO is better than SCA, PSO, MFO, MVO, and ALO for function optimizations. In addition, to visually compare the performance of IALO and the other six algorithms: SCA, PSO, MFO, MVO, and ALO, the convergence curves of SCA, PSO, MFO, MVO, ALO, and IALO performed on 15 benchmark functions $F_1(x) - F_5(x), F_7(x) - F_{13}(x), F_{15}(x), F_{16}(x), F_{18}(x)$ are generated, as shown in Figure 3. From Figure 3, it can be seen that IALO has the most rapid convergence speeds to arrive at the minimum function value.

To sum up, the proposed algorithm IALO outperforms other compared algorithms: SCA, PSO, MFO, MVO, and ALO. The results verify the performance of the IALO algorithm in solving various benchmark functions and the proposed algorithm IALO is valid.

4.2. Prediction of Chinese Influenza Based on Optimized Neural Network

4.2.1. Data Sources. Influenza is an acute respiratory infection caused by influenza virus and is also a highly infectious and fast-spreading disease. It is mainly transmitted through droplets in the air, contact between people, or contact with contaminated substances. The number of influenza patients and the incidence of influenza are determined to make hospitals supply the corresponding medical services for influenza patients. Therefore, it is essential to predict the number of influenza patients and the incidence of influenza every year.

Table 2: The average value and the standard derivation of benchmark functions $F_1(x) - F_{23}(x)$.

		SCA	PSO	MFO	MVO	ALO	IALO
	Avg.	2.4833E-02	1.7029E-07	6.6667E+02	3.5320E-01	1.0361E-05	*5.6713E-09*
F_1	Std.	8.4745E-02	3.3987E-07	2.5371E+03	9.0486E-02	9.2902E-06	6.8327E-09
	Avg.	2.7355E-05	5.6670E+00	3.5009E+01	3.8667E-01	4.0298E+01	*2.1869E-06*
F_2	Std.	7.0204E-05	6.7889E+00	1.7179E+01	1.0953E-01	4.5819E+01	1.7256E-06
	Avg.	4.4292E+03	2.1475E+01	1.4187E+04	4.0091E+01	1.1158E+03	*8.1060E-08*
F_3	Std.	3.4067E+03	8.6854E+00	1.2322E+04	1.6256E+01	6.9499E+02	6.7920E-08
	Avg.	2.2828E+01	7.0944E-01	6.7633E+01	9.3517E-01	1.2097E+01	*1.9699E-05*
F_4	Std.	1.0885E+01	1.6423E-01	9.8845E+00	3.8576E-01	3.4911E+00	1.3800E-05
	Avg.	3.1495E+02	5.9107E+01	1.8190E+04	2.4310E+02	2.1312E+02	*6.0874E+00*
F_5	Std.	7.8295E+02	4.7783E+01	3.6553E+04	4.7649E+02	3.9933E+02	7.0979E+00
	Avg.	4.4585E+00	*2.7838E-07*	1.3334E+03	3.0566E-01	8.2847E-06	4.3582E-03
F_6	Std.	4.5527E-01	5.3494E-07	3.4577E+03	6.4923E-02	6.6377E-06	4.5995E-03
	Avg.	3.6431E-02	4.0953E+00	7.1591E+00	2.0177E-02	9.9794E-02	*2.4135E-04*
F_7	Std.	3.2811E-02	5.0086E+00	1.4622E+01	9.8788E-03	2.9363E-02	2.2411E-04
	Avg.	2.2980E+00	8.3949E+01	4.2009E+04	2.5972E+00	1.8502E+00	*3.5681E-01*
F_8	Std.	2.9273E+00	1.2624E+02	7.8237E+04	2.8904E+00	2.0096E+00	1.5848E-01
	Avg.	1.4830E-02	1.0333E+02	5.0351E+02	3.8931E-01	1.4806E+00	*6.5564E-10*
F_9	Std.	4.1114E-02	1.4016E+02	5.7668E+02	3.5894E-01	2.4219E+00	9.0072E-10
	Avg.	-3.9223E+03	-7.0089E+03	-8.7739E+03	-7.7335E+03	-5.6870E+03	*-1.2465E+04*
F_{10}	Std.	2.0066E+02	7.4483E+02	1.0097E+03	7.5803E+02	7.2459E+02	3.1132E+02
	Avg.	2.3813E+01	8.5902E+01	1.6498E+02	1.0670E+02	7.9663E+01	*4.2863E-09*
F_{11}	Std.	2.9957E+01	3.3029E+01	3.3619E+01	2.7246E+01	2.0180E+01	4.8343E-09
	Avg.	1.2218E+01	1.1481E-01	1.3742E+01	1.1243E+00	2.1638E+00	*2.1607E-05*
F_{12}	Std.	9.6075E+00	3.5248E-01	8.1575E+00	6.8869E-01	4.8964E-01	1.6013E-05
	Avg.	3.0681E-01	1.1976E-02	2.7139E+01	5.7658E-01	1.4141E-02	*1.0333E-08*
F_{13}	Std.	2.5543E-01	1.4464E-02	5.3798E+01	8.2168E-02	1.2955E-02	1.0276E-08
	Avg.	3.2960E+00	*2.6382E-09*	7.6727E-01	1.4587E+00	1.0616E+01	1.3128E-03
F_{14}	Std.	6.3376E+00	5.2924E-09	1.0428E+00	9.7475E-01	4.4172E+00	1.7484E-03
	Avg.	2.2449E+03	*4.0291E-03*	1.3669E+07	6.6053E-02	1.5288E+00	2.0314E-02
F_{15}	Std.	1.2265E+04	5.3850E-03	7.4867E+07	3.3510E-02	8.0972E+00	1.9561E-02
	Avg.	1.4802E-261	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	*-9.9996E-01*
F_{16}	Std.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	3.8017E-05
	Avg.	1.5607E+00	3.7235E+00	2.8078E+00	*9.9800E-01*	1.5594E+00	1.4625E+00
F_{17}	Std.	8.9065E-01	2.7626E+00	2.2124E+00	7.7165E-12	1.0912E+00	9.6225E-01
	Avg.	9.3314E-04	3.1081E-03	1.7669E-03	4.0776E-03	4.7203E-03	*3.5287E-04*
F_{18}	Std.	3.8535E-04	5.8663E-03	3.5356E-03	7.4136E-03	7.9585E-03	3.4786E-05
	Avg.	-1.0316E+00	*-1.0316E+00*	*-1.0316E+00*	-1.0316E+00	-1.0316E+00	-1.0316E+00
F_{19}	Std.	2.1357E-05	0.0000E+00	0.0000E+00	5.7709E-08	8.1190E-14	1.3166E-04
	Avg.	3.9870E-01	*3.9789E-01*	*3.9789E-01*	3.9789E-01	3.9789E-01	4.0284E-01
F_{20}	Std.	9.6371E-04	0.0000E+00	0.0000E+00	1.5989E-07	4.0846E-14	7.3331E-03
	Avg.	3.0000E+00	*3.0000E+00*	3.0000E+00	3.0000E+00	3.0000E+00	3.0295E+00
F_{21}	Std.	1.0124E-05	0.0000E+00	4.8014E-15	4.8579E-07	2.8150E-13	3.3891E-02
	Avg.	-3.8551E+00	-3.8625E+00	*-3.8628E+00*	-3.8628E+00	-3.8628E+00	-3.8541E+00
F_{22}	Std.	2.6592E-03	1.4390E-03	2.7101E-15	4.9816E-07	2.9913E-14	1.4000E-02
	Avg.	-2.8128E+00	-3.2222E+00	*-3.2220E+00*	-3.2582E+00	-3.2705E+00	-3.1590E+00
F_{23}	Std.	4.8513E-01	9.4865E-02	5.3410E-02	6.0678E-02	5.9929E-02	1.0387E-01

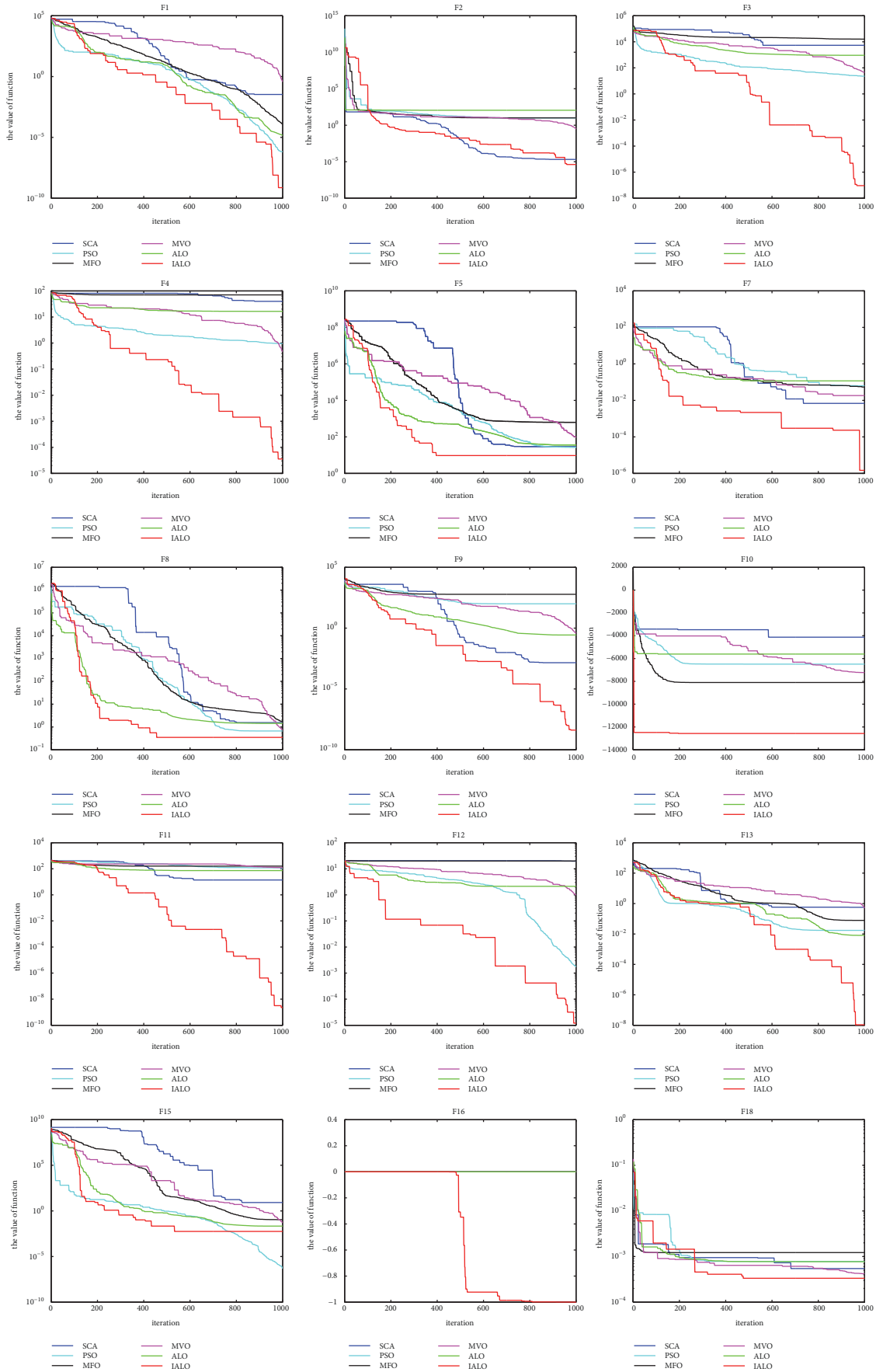

Figure 3: The convergence curves of some functions.

Table 3: Predicting the number of influenza patients.

Actual	7893	11309	14254	21953	38319
BPNN	3371	10322	14614	*22982*	8202
SCA-BPNN	12258	10556	15551	30951	32987
PSO-BPNN	473	*11454*	17101	28559	55681
MFO-BPNN	11711	11597	15070	31986	25361
MVO-BPNN	12661	9816	*13931*	31863	49451
ALO-BPNN	3794	12461	17566	27023	28565
IALO-BPNN	*10045*	9117	15643	33590	*35124*

Here, we adopt the influenza data of the whole China from January, 2004 to December, 2016 for prediction from the website http://www.phsciencedata.cn/Share/ky_sjml.jsp. From the loaded dataset, we can obtain the number of influenza patients, the number of deaths, the incidence of influenza, and the mortality rate on influenza of every month. In this paper, the number of influenza patients and the incidence of influenza are utilized to perform the predictions. Therefore, the supported data consists of number of influenza patients and the incidence of influenza from the website http://www.phsciencedata.cn/Share/ky_sjml.jsp.

4.2.2. BP Neural Network Optimized by IALO. We use IALO algorithm to optimize the weights and basis of BP neural network to create the predicted model, written as IALO-BPNN. In IALO-BPNN, every ant or antlion is mapped into the parameters: weights and basis of BP neural network. Therefore, the dimension of every ant or antlion is determined. For comparison, SCA, PSO, MFO, MVO, and ALO are also employed to optimize the weights and basis of BP neural network, and the corresponding models are SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN. And the fitness function in SCA, PSO, MFO, MVO, ALO, and IALO algorithms is defined as follows:

$$fitness = \sum_{i=1}^{Q} \left| y_i - \widehat{y}_i \right|, \tag{23}$$

where Q is the number of the data, y_i is the actual value, and $\widehat{y}_i$ is the predicted value. *For convenience, we call BPNN, SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, ALO-BPNN, and IALO-BPNN as model 1, model 2, model 3, model 4, model 5, model 6, and model 7, respectively*. We use the four evaluation criterions for models: mean absolute error (MAE), mean squared error (MSE), relative mean squared error (RMSE), and mean absolute percentage error (MAPE). The formulas of MAE, MSE, RMSE, and MAPE are defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - \widehat{y}_i \right|, \tag{24}$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \widehat{y}_i \right)^2, \tag{25}$$

Table 4: Predicting the incidence of influenza.

Actual	0.5793	0.8300	1.0462	1.6113	2.8125
BPNN	0.9826	0.8887	*1.0951*	3.7006	3.4527
SCA-BPNN	0.8857	0.7253	1.1556	2.2747	3.4892
PSO-BPNN	1.0597	0.7585	1.3530	2.3815	2.3470
MFO-BPNN	1.1822	0.8733	0.8771	2.4030	2.4213
MVO-BPNN	0.3419	0.7514	0.9569	2.7613	*2.6413*
ALO-BPNN	0.2586	*0.8221*	1.2174	2.4559	2.5905
IALO-BPNN	*0.7879*	0.8667	1.1750	*2.1211*	2.0030

$$RMSE = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \widehat{y}_i}{y_i} \right)^2, \tag{26}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \widehat{y}_i}{y_i} \right| \times 100, \tag{27}$$

where N is the number of data and y_i and $\widehat{y}_i$ denote the actual value and the predicted value of the i-th data.

4.2.3. Experimental Results on Influenza Prediction. We use the first three days' influenza data to predict the fourth day's influenza data. We choose the influenza data from January, 2004 to June, 2016 as the trained data and the influenza data from July, 2016 to December, 2016 as the tested data.

We perform model 1 to model 7 for predicting the incidence of influenza and the number of influenza patients. In model 1 and the BPNN parts of model 2 to model 7, the number of the nodes in the hidden layer is set to be 10 and the number of iterations is set to be 5000. In these models, SCA, PSO, MFO, MVO, ALO, and IALO algorithms are performed 500 iterations to optimize the weights and biases of BP neural network.

These 7 models are run only 1 time and the predicted values of the number of influenza patients and the incidence of influenza are shown in Tables 3 and 4, respectively. It is shown that the predicted model IALO-BPNN is better than the other predicted models: BPNN, SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN.

Further, we run these 7 models 10 times independently, respectively. The average MAE, MSE, RMSE, and MAPE of predicting the incidence of influenza and the average MAE, MSE, RMSE, and MAPE of predicting the number of

Table 5: MAE, RMSE, and MAPE (%) of predicting the incidence of influenza.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
MAE	0.4950	0.3615	0.5333	0.3639	0.3620	0.3984	*0.3254*
MSE	0.5869	0.3098	0.5902	0.2686	0.2586	0.3327	*0.2158*
RMSE	0.1899	0.1419	0.2761	0.1687	0.1592	0.1940	*0.0925*
MAPE	32.7250	25.7252	39.7732	28.9608	28.3542	30.5593	*23.4146*

Table 6: MAE, MSE, RMSE, and MAPE (%) of predicting the number of influenza patients.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
MAE	7448.0414	5746.8041	5968.9764	5748.0832	*4056.4442*	4648.9527	4659.0447
MSE	1.3140E+08	6.4134E+07	6.7473E+07	6.6312E+07	*3.0765E+07*	4.2187E+07	4.9102E+07
RMSE	0.3424	0.2004	0.2291	0.1939	0.1427	0.1504	*0.1050*
MAPE	40.4356	32.0196	34.8283	31.8759	25.1653	27.4949	*24.6722*

influenza patients are obtained, as shown in Tables 5 and 6, respectively.

From Table 5, it is observed that the average MAE, MSE, RMSE, and MAPE of predicting the incidence of influenza obtained by IALO-BPNN are the least and arrive at 0.3254, 0.2158, 0.0925, and 23.4146%, respectively. From Table 6, it is shown that the average RMSE and MAPE of predicting the number of influenza patients obtained by IALO-BPNN are the least and arrive at 0.1050 and 24.6722%, respectively. And the average MAE and the average MSE of predicting the number of influenza patients obtained by MVO-BPNN are the least and arrive at 4056.4442, 3.0765E+07, respectively. Therefore, it can be seen that the predicted model IALO-BPNN outperforms the other predicted models: BPNN, SCA-BPNN, PSO-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN. And the results show that IALO algorithm can effectively be used to optimize the weights and basis of BP neural network for predicting the influenza.

5. Discussion

From the results obtained from the previous sections, we can recognize that the proposed IALO in this paper shows the superior results for multidimensional functions $F_1(x)$–$F_{16}(x)$ and functions $F_{17}(x) - F_{23}(x)$ with the fixed dimension by comparison with five kinds of swarm intelligence algorithms: SCA, PSO, MFO, MVO, and ALO in Section 4.1. And IALO is applied to optimize the parameters of BP neural network and then the predicted model IALO-BPNN is built. IALO-BPNN has better predicted results than the other predicted models: BPNN, SCA-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN in Section 4.2.

The elitism operator of ALO is inspired by PSO to build the elitism operator of the proposed IALO, as shown in (20). And the improved elitism operator reveals an immediate impact on the capabilities of IALO in balancing the exploration and exploitation in dealing with function optimizations and influenza prediction. In (20), there are several parameters: inertia weight ω, accelerated factors c_1, c_2. In the experiments, we take $\omega = e^{-t/T}$ which is a decreasing function and $c_1 = c_2 = 2$. But the inertia weight ω can be selected by means of different methods, which leads to obtaining the different optimal solution and different convergence curve in optimization problems and the different predicted results in predicted problems. In addition, BP neural network in this paper is used to be optimized by IALO, but there are many kinds of neural networks. Therefore, if BP neural network is replaced by another different neural network, the results will be changed possibly.

6. Conclusion and Future Direction

In this paper, inspired by PSO, the elitism operator in ALO is improved and the improved ALO is created, written as IALO. IALO has been shown to be suitable for function optimization and influenza prediction. We use IALO to conduct the numerical tests on 23 classic benchmark functions to examine the exploration, exploitation, and convergence behavior of the proposed IALO, whose effectiveness is valid by comparison with SCA, PSO, MFO, MVO, and ALO. The comparative results illustrate that the proposed IALO is superior to SCA, PSO, MFO, MVO, and ALO. The similar results can be seen from the convergence curves. To further confirm IALO's optimization capability, we use IALO to optimize the weights and basis of BP neural network for predicting the incidence of influenza and the number of influenza patients, respectively. Thus the predicted model IALO-BPNN is built. Then IALO-BPNN is compared with the other predicted models: BPNN, SCA-BPNN, MFO-BPNN, MVO-BPNN, and ALO-BPNN. The experimental results show that the proposed model IALO-BPNN has the least MAE, MSE, RMSE, and MAPE for predicting the incidence of influenza and the least RMSE and MAPE for predicting the number of influenza patients. Therefore, there are reasons that the proposed IALO can be used to be a powerful tool in dealing with the classic benchmark functions and the combination with artificial neural network for prediction and classification.

Many worth swarm intelligence algorithm which needs to be explored in future will be proposed constantly. For example, IALO can be combined with one or more of the other swarm intelligence algorithms to become the new hybrid algorithm to further improve its performance. In addition, more than two swarm intelligence algorithms are combined to create the hybrid algorithms. These improved

hybrid algorithms can be utilized to realize the function optimizations and further able to solve the real-world problems in valid.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no financial conflicts of interest.

Acknowledgments

This work was supported by Shanxi Natural Science Foundation [grant numbers 201801D121026, 201701D121012, 201801D121008, 201701D221121]; the National Natural Science Foundation of China [grant numbers 61774137, 11571324]; the Fund for Shanxi "1331KIRT"; and Shanxi Scholarship Council of China [grant number 2016-088].

References

[1] J. H. Holland, *Adaptation in Nature and Artificial Systems*, The MIT Press, Cambridge, Mass, USA, 1992.

[2] X. Yao, *Evolutionary Computation: Theory and Applications*, World Scientific Publishing, Singapore, 1999.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[4] H. Sang, L. Gao, and Q. Pan, "Discrete artificial bee colony algorithm for lot-streaming flowshop with total flowtime minimization," *Chinese Journal of Mechanical Engineering*, vol. 25, no. 5, pp. 990–1000, 2012.

[5] S. K. Nseef, S. Abdullah, A. Turky, and G. Kendall, "An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems," *Knowledge-Based Systems*, vol. 104, pp. 14–23, 2016.

[6] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang, "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables," *Computers & Structures*, vol. 165, pp. 59–75, 2016.

[7] G. Bartsch, A. P. Mitra, S. A. Mitra et al., "Use of Artificial Intelligence and Machine Learning Algorithms with Gene Expression Profiling to Predict Recurrent Nonmuscle Invasive Urothelial Carcinoma of the Bladder," *The Journal of Urology*, vol. 195, no. 2, pp. 493–498, 2016.

[8] C.-M. Lai, W.-C. Yeh, and Y.-C. Huang, "Entropic simplified swarm optimization for the task assignment problem," *Applied Soft Computing*, vol. 58, pp. 115–127, 2017.

[9] J. H. Holl, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[10] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of IEEE Inter Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, 1995.

[11] A. V. Mokshin, V. V. Mokshin, and L. M. Sharnin, "Adaptive genetic algorithms used to analyze behavior of complex system," *Communications in Nonlinear Science and Numerical Simulation*, vol. 71, pp. 174–186, 2019.

[12] J. Fernández, J. López-Campos, A. Segade, and J. Vilán, "A genetic algorithm for the characterization of hyperelastic materials," *Applied Mathematics and Computation*, vol. 329, pp. 239–250, 2018.

[13] V. Sanchez-Tembleque, V. Vedia, L. M. Fraile, S. Ritt, and J. M. Udias, "Optimizing time-pickup algorithms in radiation detectors with a genetic algorithm," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 927, pp. 54–62, 2019.

[14] Y. M. Ding, W. L. Zhang, L. Yu, and K. H. Lu, "The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis," *Energy*, vol. 176, pp. 582–588, 2019.

[15] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[16] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[17] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[19] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Applied Mathematical Modelling*, vol. 73, pp. 109–123, 2019.

[20] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[21] M. Raju, L. C. Saikia, and N. Sinha, "Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 52–63, 2016.

[22] S. K. Majhi and S. Biswal, "Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer," *Karbala International Journal of Modern Science*, vol. 4, no. 4, pp. 347–360, 2018.

[23] Z. Wu, D. Yu, and X. Kang, "Parameter identification of photovoltaic cell model based on improved ant lion optimizer," *Energy Conversion and Management*, vol. 151, pp. 107–115, 2017.

[24] S. Mouassa, T. Bouktir, and A. Salhi, "Ant lion optimizer for solving optimal reactive power dispatch problem in power systems," *Engineering Science and Technology, an International Journal*, vol. 20, no. 3, pp. 885–895, 2017.

[25] H. L. Chen, S. Jiao, A. A. Heidari, M. J. Wang, X. Chen, and X. H. Zhao, "An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models," *Energy Conversion and Management*, vol. 195, pp. 927–942, 2019.

[26] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[27] M. Taradeh, M. Mafarja, A. A. Heidari et al., "An evolutionary gravitational search-based feature selection," *Information Sciences*, vol. 497, pp. 219–239, 2019.

[28] M. Mafarja, I. Aljarah, A. A. Heidari et al., "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Systems*, vol. 145, pp. 25–45, 2018.

[29] M. Mafarja, I. Aljarah, A. A. Heidari et al., "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowledge-Based Systems*, vol. 161, pp. 185–204, 2018.

[30] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, "Asynchronous accelerating multi-leader salp chains for feature selection," *Applied Soft Computing*, vol. 71, pp. 964–979, 2018.

[31] H. P. Hu, L. Tang, S. H. Zhang, and H. Y. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188–195, 2018.

[32] M. Qiu and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *PLoS ONE*, vol. 11, no. 5, Article ID e0155133, 2016.

[33] J. Lu, H. Hu, and Y. Bai, "Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm," *Neurocomputing*, vol. 152, pp. 305–315, 2015.

[34] J. Lu, H. Hu, and Y. Bai, "Radial basis function neural nerwork based on an improved exponential decreasing inertia weight-particle swarm optimizatin algorithm for AQI prediction," *Abstract and Applied Analysis*, vol. 2014, Article ID 178313, 9 pages, 2014.

[35] H. Hu, H. Wang, F. Wang, D. Langley, A. Avram, and M. Liu, "Prediction of influenza-like illness based on the improved artificial tree algorithm and artificial neural network," *Scientific Reports*, vol. 8, no. 1, article no. 4895, 2018.

[36] H. Hu, H. Wang, Y. Bai, and M. Liu, "Determination of endometrial carcinoma with gene expression based on optimized Elman neural network," *Applied Mathematics and Computation*, vol. 341, pp. 204–214, 2019.

[37] Y. Xu, H. Chen, A. A. Heidari et al., "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Systems with Applications*, vol. 129, pp. 135–155, 2019.

[38] A. A. Heidari, R. Ali Abbaspour, and H. Chen, "Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training," *Applied Soft Computing*, vol. 81, article 105521, 2019.