



*Research article*

## Teaching learning-based whale optimization algorithm for multi-layer perceptron neural network training

Yongquan Zhou<sup>1,2,3,\*</sup>, Yanbiao Niu<sup>1,2</sup>, Qifang Luo<sup>1,2,3</sup> and Ming Jiang<sup>4</sup>

<sup>1</sup> College of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, China

<sup>2</sup> Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

<sup>3</sup> Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning 530006, China

<sup>4</sup> Guangxi Institute of Digital Technology, Nanning 530000, China

\* **Correspondence:** Email: [yongquanzhou@126.com](mailto:yongquanzhou@126.com); Tel: +8613607882594; Fax: +867713260264.

**Abstract:** This paper presents an improved teaching learning-based whale optimization algorithm (TSWOA) used the simplex method. First of all, the combination of WOA algorithm and teaching learning-based algorithm not only achieves a better balance between exploration and exploitation of WOA, but also makes whales have self-learning ability from the biological background, and greatly enriches the theory of the original WOA algorithm. Secondly, the WOA algorithm adds the simplex method to optimize the current worst unit, averting the agents to search at the boundary, and increasing the convergence accuracy and speed of the algorithm. To evaluate the performance of the improved algorithm, the TSWOA algorithm is employed to train the multi-layer perceptron (MLP) neural network. It is a difficult thing to propose a well-pleasing and valid algorithm to optimize the multi-layer perceptron neural network. Fifteen different data sets were selected from the UCI machine learning knowledge and the statistical results were compared with GOA, GSO, SSO, FPA, GA and WOA, severally. The statistical results display that better performance of TSWOA compared to WOA and several well-established algorithms for training multi-layer perceptron neural networks.

**Keywords:** teaching learning-based; whale optimization algorithm; multi-layer perceptron (MLP) neural network; metaheuristic algorithm

---

## 1. Introduction

Since McCulloch and Pitts first proposed the primary theory and learning model of artificial neural networks [1]. ANNs can be considered as an effective model that can get desirable results for processing supervised/unsupervised machine learning tasks [2]. Hence, ANNs have been widely used in various fields as a powerful and practical tool to solve real problems, such as forecast and estimation [3], biology and medicine [4], classification [5], pattern recognition [6] and so on [7]. Among different artificial neural networks, feed-forward neural networks (FNNs) [8], especially two-layer FNNs, is widely applied [9]. In fact, FNNs with two layers is the simplest and most widespread practical application. For any type of neural network, learning is considered an important and critical process and attracts lots of researchers. The learning process is a key field and the purpose of which is to optimize the cost function for feedforward neural networks. The cost function is considered to be the mean square error (MSE) obtained by the optimal connection of the weights and bias values.

Generally, the training methods of feedforward neural networks can be seen in two classes: gradient-based search methods and metaheuristic search methods. The most popular search method in this area is the so-called Back-Propagation (BP) algorithm [10], which is a gradient-based method. When dealing with complex optimal problems, it may quickly converge to the local optimal solution instead of the global optimal solution. As a result, back-propagation cannot obtain a better candidate solution. In addition, the initial solutions of weights, biases, and learning rates determines the performance of the BP algorithm. Even if the learning target is reached, this results in the slow performance of the BP algorithm.

In recent years, a growing number of scholars' regard metaheuristic search methods as the training method of feedforward neural networks to alleviate this main drawback. In the stochastic algorithms, Montana and Davis proposed genetic algorithm (GA) is the first training method to optimize the feedforward neural network [11], they describe a set of experiments performed on data from a sonar image classification problem. The experimental results display that GA is superior to the BP algorithm when solving complex real problems. Li et al. proposed an improving particle swarm optimization algorithm based on neighborhood and historical memory (PSONHM) [12], and employed to investigate the efficiencies of PSONHM in training Multi-Layer Perceptron (MLP). The experimental results indicate that the proposed PSONHM can effectively solve the global optimization problems. Gambhir et al. developed a PSO-ANN based diagnostic model for earlier diagnosis of dengue fever [13]. In the proposed model, PSO technique is applied to optimize the weight and bias parameters of ANN method. The effectiveness of the proposed model is evaluated based on accuracy, sensitivity, specificity, error rate and AUC parameters. The results of the proposed model have been compared with other existing approaches like ANN, DT, NB, and PSO. Mirjalili et al. proposed using a social spider optimization algorithm to train feedforward neural networks [14]. The experimental results show that the algorithm has fast convergence speed and high accuracy in most test data sets. Uzlu et al. study is associated with predicting energy consumption in Turkey [15]. GDP (gross domestic product), population, import and export were used as predictor variables. TLBO (teaching-learning-based optimization) and BP were used to train ANNs. ANN-TLBO predicted the energy consumption more accurately than ANN-BP. Using the ANN-TLBO model, the energy consumption was forecasted until 2020. Aljarah et al. proposed a

new training algorithm based on the recently proposed whale optimization algorithm (WOA) [16]. The results are verified by comparisons with back-propagation algorithm and six evolutionary techniques. Kowalski and Łukasik to compare the Krill Herd Algorithm (KHA) optimization algorithm used for learning an artificial neural network (ANN), with other heuristic methods and with more conventional procedures [17]. The proposed ANN training method has been verified for the classification task. It has been concluded that the application of KHA offers promising performance-both in terms of aforementioned metrics, as well as time needed for ANN training. Alboaneen et al. used to glowworm swarm optimisation (GSO) algorithm training the MLP neural network [18]. The GSO based trainer is evaluated on five classification datasets, the results show that our proposed trainer achieves better classification accuracy rate in most datasets compared to the other algorithms. Heidari et al proposed a new hybrid stochastic training algorithm using the recently proposed grasshopper optimization algorithm (GOA) for multilayer perceptrons (MLPs) neural networks [19]. The proposed GOAMLP model is then applied to five important datasets: breast cancer, parkinson, diabetes, coronary heart disease, and orthopedic patients. It is shown and proved that the proposed stochastic training algorithm GOAMLP is substantially beneficial in improving the classification rate of MLPs. Ehsan, et al. proposed an improved cuckoo search algorithm [20]. It is employed for training feedforward neural networks for two benchmark classification problems. Socha and Blum proposed a Ant Colony Optimization (ACO) variant for continuous optimization [21,22]. The results show that the best of our algorithms are comparable to gradient-based algorithms for neural network training [23–26].

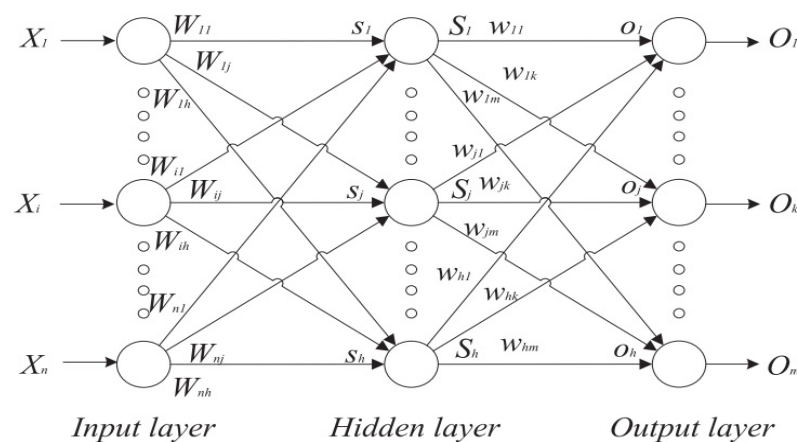
In 2016, Mirjalili and Lewis proposed the whale optimization algorithm [27]. It has proved that the WOA algorithm is more competitive than other famous meta-heuristic algorithms. When dealing with the real problem, it may quickly reach to the local optimal solution instead of the global optimal solution. This phenomenon, called immature convergence, is a common problem in metaheuristics [28]. Another question called the stagnation in local optimal solutions [29]. This question can seriously lead to the degradation of the candidate solution quality. The phenomenon can happen when algorithms fail to get better between exploration and exploitation. Exploration defined as the ability of algorithm to find an unknown area in search space. The exploitation stage must follow the exploration phase. In this stage, the field of the better-explored positions in various regions is included again to get a better candidate solution [30]. Widespread exploration in early iterations and then focused exploitation in the final steps of optimization can help the optimizer to avert local solutions [31].

Studies have shown that some algorithms have better local search capabilities [32], some algorithms have better global search capabilities, and some optimization algorithms profit from a better balance between exploration and exploitation. Hence, hybrid algorithms can decrease the inherent defects of the original algorithm [33]. With this strategy, the hybrid algorithms gain all the superiority of the original algorithm. The primary intention of hybrid optimizers is to decrease the amount of computation, increase the accuracy of the results, enhanced the stability of the algorithm and the convergence trend of the basic algorithm [34]. This paper presents an improved whale optimization algorithm which aimed to enhance the precision of the convergence of the basic WOA algorithm. In order to test the performance of the algorithm, the optimal weight and biases of MLP neural networks are used for classification problems and experimental results indicate that this algorithm has better classification accuracy, convergence speed, and convergence precision than other algorithms in optimizing multi-layer perceptron neural network.

The research content of this paper is as follows: the basic definition of MLP is described in section 2. Section 3 describes the original WOA algorithm. The detailed description of the TSWOA is presented in section 4. Section 5 details how the TSWOA can be used for training MLP. Various experimental results and analyses were conducted in section 6. We conclude in section 7.

## 2. Feedforward neural network and multi-layer perceptron

Feedforward neural network is a prevalent structure of neural network model which employs its sophisticated parallel layered to understand and approximate computing models [35]. It is made up of many neurons distributed in different levels. The input layer is located in the first layer of feedforward neural network, the output layer is located in the last layer, and the hidden layer is located in the other layer. Feedforward neural networks that have many hidden layers are called multilayer perceptron (MLP), which is displayed in Figure 1. In MLP, neurons are organized in a unidirectional pattern. In the network form, the message is spread in one direction, respectively through the input layer, the hidden layer, and the output layer. As shown in Figure 1, the data is input by the input layer, multiplied by the respective weights, and then taken as the input of the hidden layer. After the calculation of the hidden layer, the data is spread to the output layer by multiplying the respective weights, and the output is calculated by the output layer. The actual output of the neuron is calculated by the activation function. The operation step of MLP is displayed in Eqs (1)–(4).



**Figure 1.** Three-layer FNNs.

Firstly, the input value of the hidden layer is obtained by using the Eq (1)

$$s_j = \sum_{i=1}^n (w_{i,j} * x_i) - \theta_j, \quad j = 1, 2, \dots, h \quad (1)$$

where  $n$  represents the total number of the input layer,  $w_{i,j}$  is the connection weight, bias value of the  $j$ -th node is expressed as  $\theta_j$ , and  $x_i$  is  $i$ -th input data.

The output value of the hidden layer is obtained by using the Eq (2),

$$S_j = \text{sigmoid}(s_j) = \frac{1}{(1 + \exp(-s_j))}, \quad j = 1, 2, \dots, h \quad (2)$$

Finally, the result of the output layer is described as follows:

$$o_k = \sum_{j=1}^h (w_{j,k} * S_j) - \theta'_k, \quad k = 1, 2, \dots, m \quad (3)$$

$$O_k = \text{sigmoid}(o_k) = \frac{1}{(1 + \exp(-o_k))}, \quad k = 1, 2, \dots, k \quad (4)$$

where  $w_{j,k}$  is the connection weight from the  $j$ -th node to the  $k$ -th node,  $\theta'_k$  expresses the bias value of the  $k$ -th node in the output layer, and  $O_k$  is the output value of the  $k$ -th node. From Eqs (1)–(4), it can be concluded that the connection weights and bias values are the most important parts of the MLP, and they confirm the final value of the output. The ultimate goal of optimizing MLP is to get the optimal connection weights and bias to achieve the output.

### 3. Whale Optimization Algorithm (WOA)

There are three main stages in the whale optimization algorithm, which are searching prey, encircling prey and Logarithmic spiral prey. Among them, searching for prey is the exploration stage of the algorithm. Moreover, encircling prey and Logarithmic spiral prey is the exploitation stage of the algorithm. This method of Logarithmic spiral prey is based on creating bubbles by encircling or through a ‘9’ shaped path [27]. It can be modeled as follows:

#### 3.1. Searching prey

At this stage, the humpback whale does not know the position of the prey. The humpback whale can only change randomly to update its position. This means that humpback whales randomly select the position of an individual from the current population as prey to update its position. The humpback whales position is shown by the following Eqs (5)–(6):

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X} \right| \quad (5)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (6)$$

where  $\vec{X}_{rand}$  represents the random vector (a random whale). The coefficient vectors are represented by  $\vec{A}$  and  $\vec{C}$ , and the calculation process is expressed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (7)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (8)$$

where the vector  $\vec{a}$  is declining linearly from 2 to 0, and  $\vec{r}$  is a random vector in the range of 0 and 1.

### 3.2. Encircling prey

In the WOA, since it is impossible to know the location of the optimal solution in the search space, the algorithm assumes that the prey is the best candidate solution (the optimal whale location) or close to the optimal solution. After obtaining the best whale position, all whales update their positions towards the current optimal whale position. The calculation process shown as follows:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \quad (9)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (10)$$

where current iteration is shown by  $t$ ,  $X^*$  represents the position of the optimal solution. if  $|A| > 1$ , the searching prey is represented by Eqs (5)–(6), otherwise the encircling prey is represented by Eqs (9)–(10), which expresses the shrinking mechanism.

### 3.3. Logarithmic spiral updating position

At this stage, the distance between the individual whale and the optimal whale (prey) position is first calculated, and then the whale moves up the water surface in a spiral shape while spitting out many bubbles of varying sizes to prey. The whales are updated as follows:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (11)$$

where  $\vec{D}' = \left| \vec{X}^*(t) - \vec{X}(t) \right|$  shows the distance from whale to its prey,  $l$  is a random number, which is the range of  $-1$  and  $1$ , the shape of the logarithmic spiral is defined as  $b$ , which is the constant number.

In the process of predation, whales have two ways of predation. Therefore, we assume that their probability of encircling mechanism is 50%, and the spiral model updates their positions with the same probability, and its model is as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p > 0.5 \end{cases} \quad (12)$$

where  $p$  is a random number, which is the range of 0 and 1. The pseudo-code of WOA can be displayed in Algorithm 1 below.

---

#### Algorithm 1. WOA pseudo-code

---

1. Initialize the whale population  $X_i (i = 1, 2, \dots, n)$
  2. Initialize  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  3. Calculate the fitness of each search agent
  4.  $X^*$  = the best search agent
  5. **while** ( $t < \text{max\_iteration}$ )
  6. **for** each search agent
  7.     Update  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  8. **if1** ( $p < 0.5$ )
  9. **if2** ( $|A| < 1$ )
-

- 
10. Update the position of the current search agent by Eq (10)
  11. **else if2** ( $|A| \geq 1$ )
  12.     Select a random search agent ( $X_{rand}$ )
  13.     Update the position of the current search agent by Eq (6)
  14. **end if2**
  15. **else if1** ( $p \geq 0.5$ )
  16.     Update the position of the current search by Eq (11)
  17. **end if1**
  18. **end for**
  19.     Check if any search agent goes beyond the search space and amend it
  20.     Calculate the fitness of each search agent
  21.     Update  $X^*$  if there is a better solution
  22.      $t = t + 1$
  23. **end while**
  24. Return  $X^*$
- 

#### 4. An improved teaching learning-based whale optimization algorithm (TSWOA)

In the literature [36], the effectiveness of WOA in solving some real problems has been verified. Due to the role of leaders, the WOA has better search capability. These random leaders enable the whale algorithm to increase the diversity of the population in the early stages. However, the algorithm has problems such as slow convergence accuracy and difficulty in jumping out of local optimum. In addition, when researchers use WOA to solve complex real-world optimization problems, WOA needs some improvements to obtain better performance. Inspired by the characteristics of the teaching-learning-based optimization algorithm, we tried to improve the search ability of the WOA algorithm by utilizing the advantages of teaching-learning-based optimization algorithm (TLBO) [37].

It is called TSWOA that the teaching phase of TLBO is added to the whale optimization algorithm. On the one hand, it enhances the exploitation capability of the algorithm and improves the quality of candidate solutions. On the other hand, from the biological background, whales are considered as highly intelligent animals with thinking, learning, judging, and communication. TSWOA enables whales to have the ability of self-learning, which makes the algorithm more consistent with the biological mechanism. Meanwhile, the improved algorithm is more self-organizing and cooperative. We add the simplex method to the WOA to increase the diversity of the population and enrich the exploration capacity of the algorithm. The description of the new method is provided in the next subsections.

##### 4.1. Teaching learning-based optimization (TLBO)

TLBO is a metaheuristic algorithm whose idea comes from the traditional learning process to simulate the teaching of teachers and the learning of students in the class to achieve the purpose of optimization [37,47]. A class is a population in the search space, and the number of people in the class is the size of the population. The subject the student learns could be interpreted as different design variables. The learning level represents the fitness of the optimization algorithm, and the teacher is the individual with the best fitness value of the population. There are two mechanisms for

exchanging information in the classroom: collaboration between teachers and students and between students. These intra-class actions implemented in two different consecutive processes: the teacher phase is to simulate the influence of teachers on students, and the learner phase is to model the cooperative learning between students.

#### 4.1.1. Teacher phase

In the teacher phase, the students with the best fitness values are selected as teachers to improve the learning level of other students in the class. The average level of the class continues to improve, moving towards the optimal solution. The difference between the average level of students (population mean) and teachers is as follows:

$$D_{\text{difference}} = \text{rand}(0,1)(X_{\text{teacher}} - T_f X_{\text{mean}}) \quad (13)$$

where  $X_{\text{teacher}}$  is the individual with the best fitness value of the current population.  $X_{\text{mean}}$  is the mean value of the class, which is defined as  $X_{\text{mean}} = \sum_{i=1}^n X^i / n$ . The value of  $T_f$  can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as  $T_f = \text{round}(1 + \text{rand}(0,1))$ . In the teacher phase, the  $i$ -th learner in the class generates a new individual according to the following:

$$X_{\text{new}}^i = X_{\text{old}}^i + D_{\text{difference}} \quad (14)$$

#### 4.1.2. Learner Phase

In the actual teaching process, the teacher's teaching is a heuristic process, and students may not be able to improve themselves effectively according to the teacher's teaching. Continuous learning from other students after class can improve their ability more effectively, which is the learning stage of the TLBO algorithm.

In the learner phase, randomly select two learners  $X_i$  and  $X_j$  from the class, where  $i \neq j$ . The  $i$ -th learner in the class generates a new individual according to the following:

$$X_{\text{new}}^i = \begin{cases} X_{\text{old}}^i + \text{rand}(0,1)(X_i - X_j) & f(X_i) < f(X_j) \\ X_{\text{old}}^i + \text{rand}(0,1)(X_j - X_i) & f(X_i) > f(X_j) \end{cases} \quad (15)$$

### 4.2. Our approach (TSWOA)

#### 4.2.1. The combination of teacher phase and whale optimization algorithm

In this section, we mainly introduce the combination of teacher phase and whale optimization algorithm and we explain how to improve the quality of candidate solutions. To increase the exploitation, avoid falling into local optimum, and improve the quality of candidate solutions, we consider adding the teacher phase of TLBO to the algorithm to enable whale agents to have a capacity of self-learning. After each update of the whale agent, the fitness value of each whale agent was calculated. Then select the best whale agent and use formula 14 to guide other whale agents to



move towards the current best position to improve the average performance of individuals in the population. The implementation steps of TWOA as follows:

---

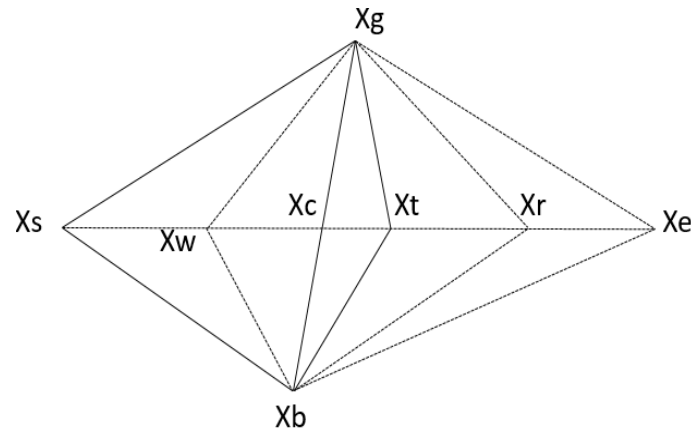
**Algorithm 2.** TWOA pseudo-code

---

1. Initialize the whales population  $X_i (i = 1, 2, \dots, n)$
  2. Initialize  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  3. Calculate the fitness of each search agent
  4.  $X^*$  = the best search agent
  5. **while** ( $t < \text{max\_iteration}$ )
  6. **for** each search agent
  7.     Update  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  8.     **if1** ( $p < 0.5$ )
  9.     **if2** ( $|A| < 1$ )
  10.         Update the position of the current search agent by Eq (10)
  11.     **else if2** ( $|A| \geq 1$ )
  12.         Select a random search agent ( $X_{rand}$ )
  13.         Update the position of the current search agent by Eq (6)
  14.     **end if2**
  15.     **else if1** ( $p \geq 0.5$ )
  16.         Update the position of the current search by Eq (11)
  17.     **end if1**
  18.     Update the position of each search agent by Eq (14)
  19.     Check if any search agent goes beyond the search space and amend it
  20. **end for**
  21.  $t = t + 1$
  22. **end while**
  23. Return  $X^*$
- 

#### 4.2.2. The Simplex Method (SM)

Although TWOA enhances the convergence precision of the WOA, it is easy to fall into the local optimum, so the Simplex method (SM) [38] is added to not only increase the diversity of population but also change the search direction of the algorithm and expand the search space. The simplex method obtains higher search accuracy by optimizing and has a good advantage in searching. The simplex method uses the optimization idea of reflection, expansion, and contraction of a convex figure (simplex) in multidimensional space to reach the minimum point [39]. Therefore, it is easy to help the algorithm to jump out of local optima. In this paper, after each update of the whale agent in the TWOA algorithm, the simplex method is run, where the specific addition position is shown in the pseudocode of Algorithm 3. The schematic diagram of the simplex method (Figure 2) is shown below:



**Figure 2.** diagrams of the Simplex method.

The specific process of the simplex method is described as follows:

**Step 1:** Obtaining the objective function value of all points, and getting the best point as the  $X_g$ , and then getting the second-best point as the  $X_b$ , supposing that  $X_s$  is the one should be substituted.  $f(X_s)$ ,  $f(X_b)$  and  $f(X_g)$  show the objective function values.

**Step 2:** Obtaining the objective function value of the middle point  $X_c$  between point  $X_g$  and point  $X_b$ :

$$X_c = \frac{X_g + X_b}{2} \quad (16)$$

**Step 3:** Obtaining the reflection point  $X_r$  by using the following formula.  $\alpha$  is the reflection coefficient, which is set as 1:

$$X_r = X_c + \alpha (X_c - X_s) \quad (17)$$

**Step 4:** If  $f(X_r) < f(X_g)$ , getting the expansion point from formula 18:

$$X_e = X_c + \gamma (X_r - X_c) \quad (18)$$

where the expansion coefficient is  $\gamma$ , which usually set to 2. If  $f(X_e) < f(X_g)$ ,  $X_s$  will be replaced by  $X_e$ ; otherwise,  $X_s$  will substitute  $X_r$ .

**Step 5:** If  $f(X_r) < f(X_s)$ , the compression point can be acquired by the formula (19):

$$X_t = X_c + \beta (X_s - X_c) \quad (19)$$

where  $\beta$  represents the cohesion coefficient, which is set to 0.5. If  $f(X_t) < f(X_s)$ , replace  $X_s$  by  $X_t$ ; otherwise,  $X_t$  substitute  $X_s$ .

**Step 6:** If  $f(X_g) < f(X_r) < f(X_s)$ , shrink point is  $X_w$ , and the shrink coefficient is  $\beta$ .

$$X_w = X_c - \beta (X_s - X_c) \quad (20)$$

If  $f(X_w) < f(X_s)$ , substitute  $X_s$  by  $X_w$ ; otherwise, substitute  $X_s$  by  $X_r$ .

According to the simplex method above, this method diversifies the population, ensuring the

algorithm can search efficiently, and reducing the algorithm fall into local optimum. The implementation steps of TSWOA are as follows.

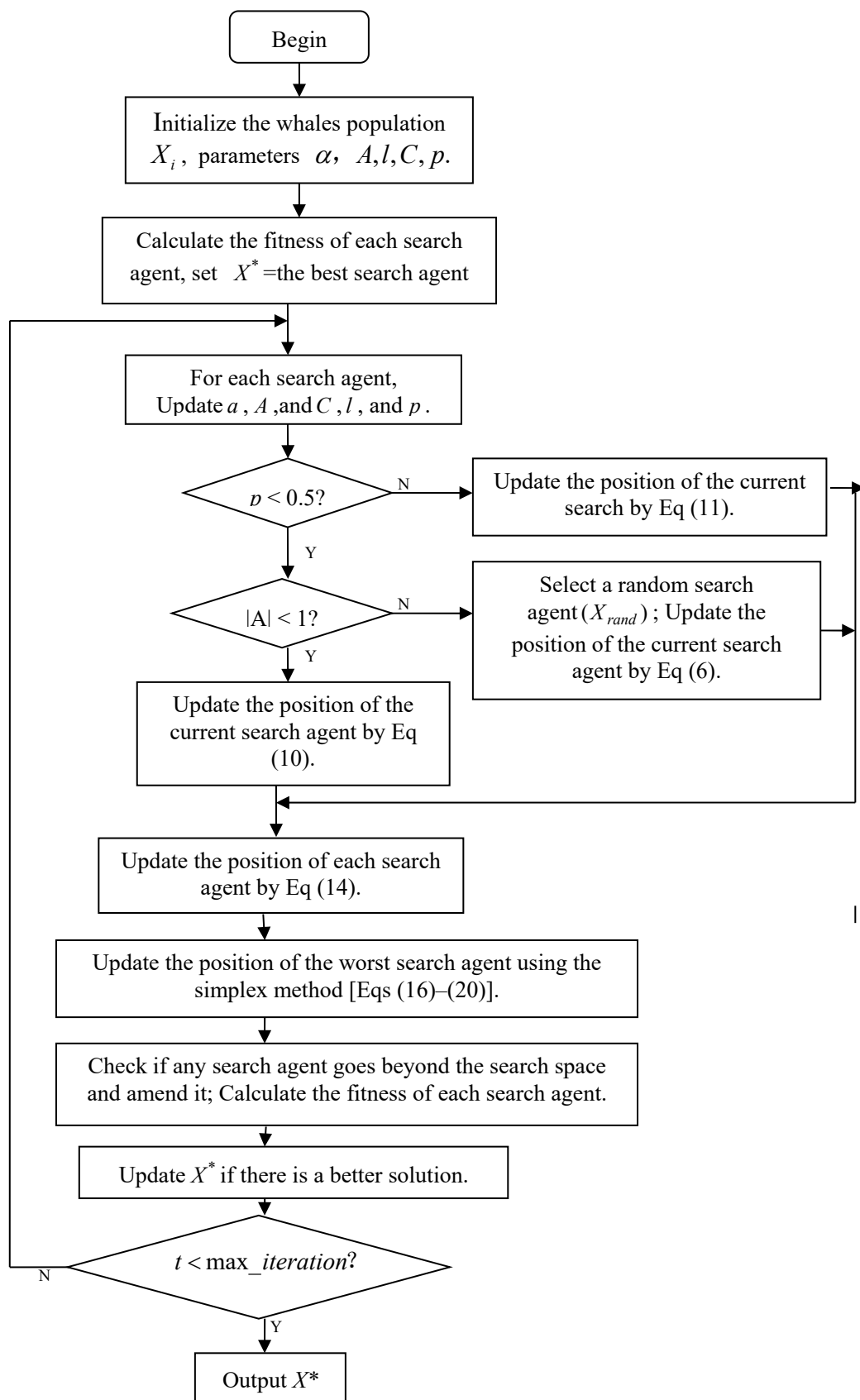
---

**Algorithm 3.** TSWOA pseudo-code

---

1. Initialize the whales population  $X_i (i = 1, 2, \dots, n)$
  2. Initialize  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  3. Calculate the fitness of each search agent
  4.  $X^*$  = the best search agent
  5. **while** ( $t < \text{max\_iteration}$ )
  6. **for** each search agent
  7.     Update  $a$ ,  $A$ , and  $C$ ,  $l$ , and  $p$
  8.     **if1** ( $p < 0.5$ )
  9.     **if2** ( $|A| < 1$ )
  10.     Update the position of the current search agent by Eq (10)
  11.     **else if2** ( $|A| \geq 1$ )
  12.     Select a random search agent ( $X_{rand}$ )
  13.     Update the position of the current search agent by Eq (6)
  14.     **end if2**
  15.     **else if1** ( $p \geq 0.5$ )
  16.     Update the position of the current search by Eq (11)
  17.     **end if1**
  18.     Update the position of each search agent by Eq (14)
  19.     Update the position of the worst search agent using the simplex method [Eqs (16)–(20)]
  20.     **end for**
  21. Check if any search agent goes beyond the search space and amend it
  22. Calculate the fitness of each search agent
  23. Update  $X^*$  if there is a better solution
  24.  $t = t + 1$
  25. **end while**
  26. Return  $X^*$
- 

The corresponding to the TSWOA flowchart as follow (see Figure 3):



**Figure 3.** TSWOA flowchart.

## 5. Using TSWOA to train MPL

In this part, we focus on ways to solve the problem. It covers the setting of the candidate solution dimension, the setting of the objective function and the setting of assessment to the experiment results.

### 5.1. The setting of the dimensionality of the candidate solution

Before optimizing MLP, we must first obtain the structure of the MLP network. The number of neurons of the input layer and the output layer is decided by classified data sets, while the number of neurons of the hidden layer is decided by the Kolmogorov theorem [40]. The result is determined by Eq (21).

$$Hidden = 2 \times Input + 1 \quad (21)$$

when TSWOA is adopted to obtain the connection weights and bias values,  $D$  represents the dimension of the candidate solution. The result is determined by Eq (22):

$$D = (Input \times Hidden) + (Hidden \times Output) + Hidden_{bias} + Output_{bias} \quad (22)$$

where the number of neurons of the input layer, hidden layer, and output layer is expressed by  $Input$ ,  $Hidden$  and  $Output$ . The number of bias of the hidden layer and the output layer is shown by  $Hidden_{bias}$  and  $Output_{bias}$ .

### 5.2. Setting of candidate solution form

In the study [41], we can get many encoding ways to optimize MPL. The mathematical models of the TSWOA algorithm are all represented by vectors, so we use vectors to represent.  $X = \{X_1, X_2, X_3, \dots, X_n\}$  represents  $N$  agents in the population. Each agent in population is denoted by  $X_i = \{i_w, h_w, h_b, o_b\}$ ,  $i = 1, 2, \dots, n$ , the weights of the input layer and the hidden layer are denoted by  $i_w$  and  $h_w$ , the biases of the hidden layer and the output layer are shown by  $h_b$  and  $o_b$ .

### 5.3. The setting of objective function

In optimizing MLP, it is crucial to define appropriate objective functions. Generally, after agents are defined by vectors, the objective function of the neural network needs to be used to define the fitness function of the TSWOA algorithm. Hence, the fitness function is considered as the difference between the theoretical output and the actual output in the neural network, which is obtained by using Eq (23):

$$MSE = \sum_{i=1}^m (o_i^k - d_i^k)^2 \quad (23)$$

where the total number of output is shown by  $m$ , the desired output and the actual output of  $i$ -th input are exhibited by  $d_i^k$  and  $O_i^k$ .

Apparently, training neural networks requires data sets, and there are many training data sets. So, the fitness function is the mean value of MSE, which is obtained by using formula 24:

$$\overline{\text{MSE}} = \sum_{k=1}^s \frac{\sum_{i=1}^m (o_i^k - d_i^k)^2}{s} \quad (24)$$

where the total number of training data sets is shown by  $S$ , other parameters are the same as the formula above. Therefore, the objective function of optimizing MLP is defined:

$$F = \text{Minimize } (\overline{\text{MSE}}) \quad (25)$$

#### 5.4. Setting of assessment

Because classified data is used to optimize MLP, classification problems often involve learning existing data to forecast undiscovered data. In classification problems, accuracy is considered to assessing the classification capacity of the algorithm, which is defined as follows:

$$\text{Accuracy} = \frac{\overline{N}}{N} \quad (26)$$

where  $\overline{N}$  shows the correct number for the classification, the total number of datasets is denoted by  $N$ .

## 6. Experiment and result analysis

In this section to test the validity of the algorithm for optimizing MPL, fifteen data sets from the UCI machine learning knowledge were obtained. There are 15 data sets displaying in Table 1. To better test the performance of TSWOA, the other six state-of-the-art algorithms are adopted: grasshopper optimization algorithm (GOA) [19], glowworm swarm optimization (GSO) [42], social spider optimization algorithm (SSO) [43], flower pollination algorithm (FPA) [19], genetic algorithm (GA) [11], and whale optimization algorithm (WOA) [44].

**Table 1.** The details of data sets.

Datasets	Attribute	Class	Training	Testing	Input	Hidden	Output
Blood	4	2	493	255	4	9	2
Scale	4	3	412	213	4	9	3
Survival	3	2	202	104	3	7	2
Liver	6	2	227	118	6	13	2
Seeds	7	3	139	71	7	15	3
Wine	13	3	117	61	13	27	3
Iris	4	3	99	51	4	9	3
Statlog	13	2	178	92	13	27	2
Cancer	9	2	461	238	9	19	2
Diabetes	8	2	507	261	8	17	2
Gene	57	2	70	36	57	115	2
Parkinson	22	2	129	66	22	45	2
Splice	60	2	660	340	60	121	2
WDBC	30	2	394	165	30	61	2
Zoo	16	7	67	34	16	33	7

### 6.1. Experimental setup

All used codes in this study have been implemented in the same manner using MATLAB R2017 (a) and run on a PC with the Windows 10 64-bit and Intel Core (TM) i5-4590 processor, 3.30 ghz and 4 GB RAM. For all data sets above, data for training and testing respectively accounted for 66% and 34% of the total data [45]. To obtain great accuracy of experimental results, each algorithm runs 20 times. The number of iterations is 500 in independent experiments. The population number is 30 and the main parameter settings for all algorithms are in Table 2.

**Table 2.** The initial parameters of algorithms.

Algorithms	Parameter values
GOA [19]	$C_{min} = 0.00004, C_{max} = 1$
GSO [42]	Luciferin decay value is 0.4, luciferin enhancement is 0.6, the rate of the neighborhood range is 0.08, the number of neighbors is 5, the step size of moving is 0.3, initial luciferin is 5
SSO [43]	$PF = 0.7$
FPA [42]	$P = 0.8$
GA [11]	Crossover rate is 1, Mutation rate is 0.01
WOA [44]	$\vec{\alpha}$ Linearly decreased from 2 to 0 have been used as recommended in [44].
TSWOA	$\vec{\alpha}$ Linearly decreased from 2 to 0 have been used as recommended in [44]

### 6.2. Experimental results and analysis of training MLP

The 15 data sets were obtained from the UCI machine learning knowledge, namely, Blood, Scale, Survival, Liver, Seeds, Wine, Iris, Statlog, Cancer, Diabetes, Gene, Parkinson, Splice, WDBC, and Zoo. The experimental results are displayed in Tables 3–17, in which the “Best” is the best value, the “Worst” is worst value, the “Mean” is the mean value and the “Std” is the standard deviation. Accuracy is the best accuracy of 20 independent runs, and rank is the algorithm ranked according to accuracy. Due to the randomness of the swarm intelligence algorithm in the process of operation, the statistical test is important [46]. In order to obtain the difference between the improved algorithm and contrastive algorithms, Wilcoxon’s rank-sum test is also added in the experiment to obtain whether there is an obvious difference between the two groups of data. When the value is less than 0.05, the difference between the two groups of data is obvious. The  $p$  values are revealed in Table 18.

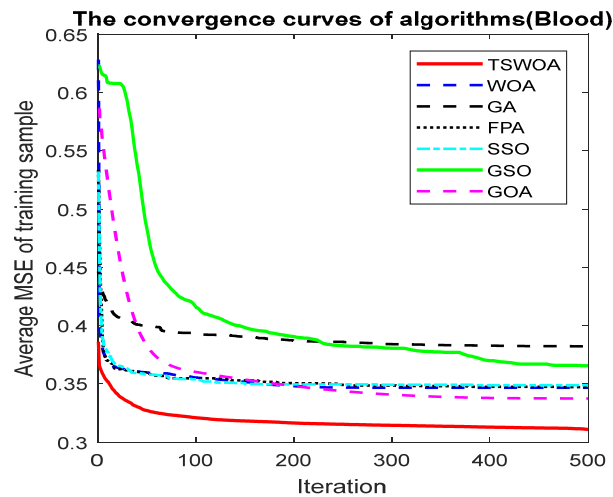
#### 6.2.1. Blood classification problem

The blood classification problem is the most commonly used data set in the classification problem. The experimental results of the classification problem of Blood dataset trained by intelligence optimization algorithm are shown in Table 3, and the convergence and the anova graphs of algorithms are displayed in Figures 4 and 5, severally. It was found from Table 3 that the mean value and variance of MSE obtained by the TSWOA algorithm are better than other comparison algorithms, indicating that the optimizing result of the TSWOA algorithm is the best. Experimental results show that the accuracy of the algorithm is 80.39, higher than that of other comparison algorithms, ranking first. As can be found from the convergence curve in Figure 4, the convergence

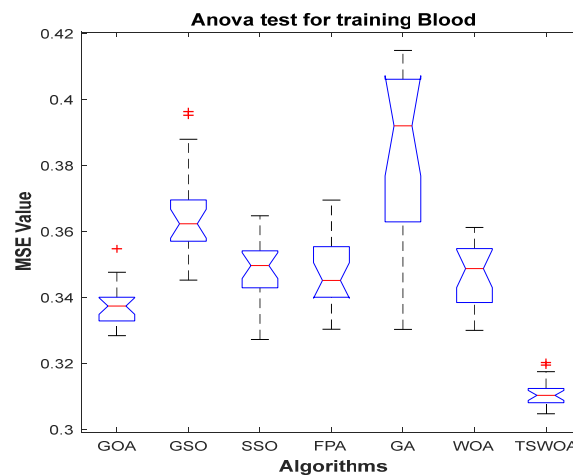
speed of the TSWOA algorithm is faster than that of other algorithms and the accuracy value is better than that of other algorithms. According to Figure 5, compared with other comparison algorithms, TSWOA has a better variance value. According to the  $P$ -value test shown in Table 18, the results of TSWOA is obviously different from other comparison algorithms.

**Table 3.** The experimental results of Blood.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	3.28E - 01	3.55E - 01	3.37E - 01	6.32E - 03	78.43	3
GSO	3.45E - 01	3.96E - 01	3.66E - 01	1.49E - 02	76.47	6
SSO	3.27E - 01	3.65E - 01	3.49E - 01	9.11E - 03	78.82	2
FPA	3.30E - 01	3.69E - 01	3.47E - 01	9.69E - 03	78.04	4
GA	3.30E - 01	4.15E - 01	3.82E - 01	2.60E - 02	77.25	5
WOA	3.30E - 01	3.61E - 01	3.47E - 01	1.06E - 02	76.25	7
TSWOA	3.05E - 01	3.20E - 01	3.11E - 01	4.37E - 03	80.39	1



**Figure 4.** The convergent curve of Blood.



**Figure 5.** The variance diagram of Blood.

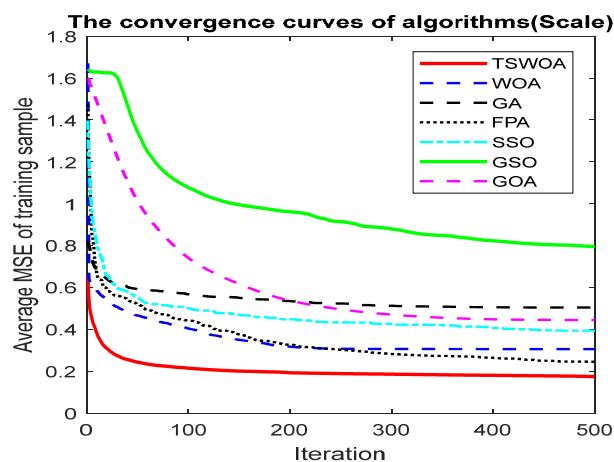


### 6.2.2. Scale classification problem

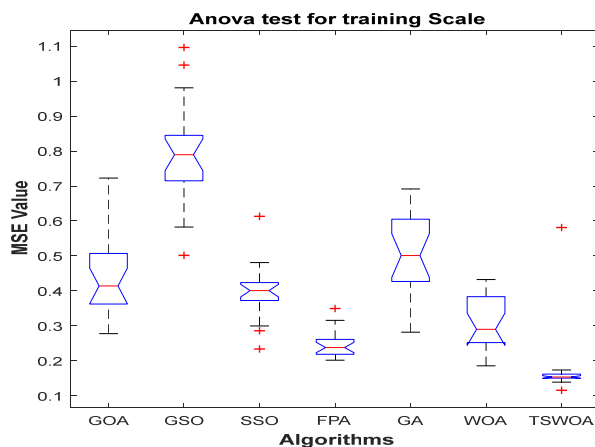
The experimental results of 20 independent runs of the algorithm are displayed in Table 4, and the convergence and anova graphs of all algorithms are displayed in Figures 6 and 7, severally. As displayed in Table 4, the mean value of MSE obtained by TSWOA algorithm is the smallest in the data set. But, the accuracy of the TSWOA algorithm and the FPA algorithm in optimizing MLP to forecast Scale data set is 89.67, ranking first, and their classification accuracy is far higher than that of other comparison algorithms. As seen in Figure 6, the TSWOA algorithm has faster convergence speed and higher convergence accuracy than other comparison algorithms. Figure 7 exhibits that TSWOA has a small and stable anova. Table 18 also shows the superiority of TSWOA, which is obviously different from all comparison algorithms in Scale dataset.

**Table 4.** The experimental results of Scale.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	2.78E - 01	7.23E - 01	4.45E - 01	1.11E - 01	78.87	3
GSO	5.02E - 01	1.10E + 00	7.95E - 01	1.42E - 01	61.50	6
SSO	2.34E - 01	6.13E - 01	3.94E - 01	7.89E - 02	77.94	4
FPA	2.01E - 01	3.50E - 01	2.46E - 01	3.73E - 02	89.67	1
GA	2.82E - 01	6.91E - 01	5.04E - 01	1.09E - 01	76.53	5
WOA	1.86E - 01	4.33E - 01	3.06E - 01	7.43E - 02	86.38	2
TSWOA	1.16E - 01	5.81E - 01	1.75E - 01	9.65E - 02	89.67	1



**Figure 6.** The convergent curve of Scale.



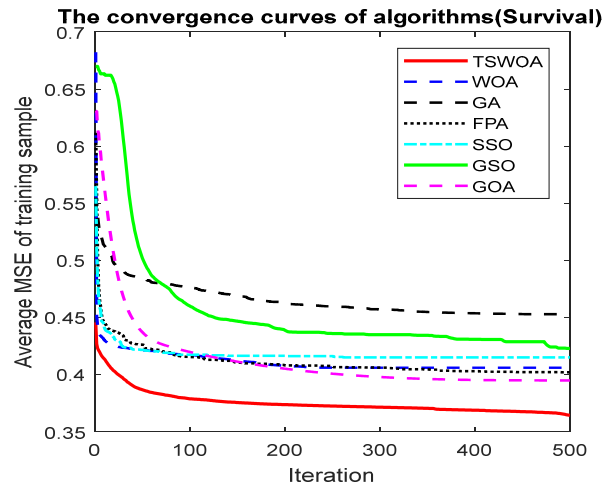
**Figure 7.** The variance diagram of Scale.

### 6.2.3. Survival classification problem

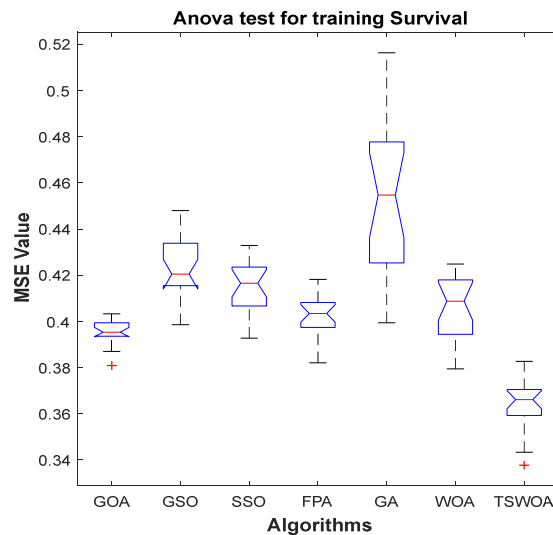
Table 5 displays the experimental results of the seven algorithms that optimized MLP to forecast the Survival dataset. The convergence and variance graphs of all algorithms are displayed in Figures 8 and 9, severally. As displayed in Table 5, the mean value of MSE obtained by the TSWOA algorithm is the smallest in the comparison algorithms. But, the accuracy of the TSWOA algorithm optimizing MLP in forecasting the Survival dataset is lower than that of GOA, GSO, SSO, and GA, which ranks third. Among them, the classification accuracy of GOA, GSO and SSO algorithms is 81.73, ranking first. The convergence curve in Figure 8 shows that the TSWOA algorithm has a faster convergence speed and a better convergence accuracy value than other algorithms. In Figure 9, compared with other algorithms, TSWOA has a better mean value. Table 18 displays that the TSWOA algorithm has obvious advantages.

**Table 5.** The experimental results of Survival.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	3.81E - 01	4.03E - 01	3.95E - 01	5.57E - 03	81.73	1
GSO	3.99E - 01	4.48E - 01	4.23E - 01	1.37E - 02	81.73	1
SSO	3.93E - 01	4.33E - 01	4.15E - 01	1.05E - 02	81.73	1
FPA	3.82E - 01	4.18E - 01	4.02E - 01	9.55E - 03	79.8	4
GA	3.99E - 01	5.16E - 01	4.53E - 01	3.16E - 02	81.72	2
WOA	3.79E - 01	4.25E - 01	4.06E - 01	1.40E - 02	81.73	1
TSWOA	3.38E - 01	3.83E - 01	3.64E - 01	1.10E - 02	80.77	3



**Figure 8.** The convergent curve of Survival.



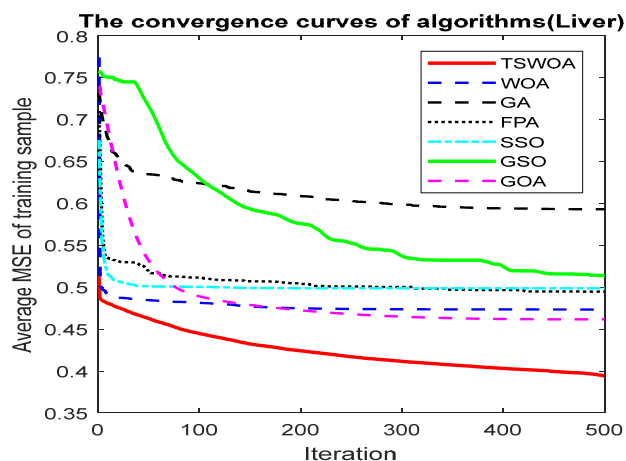
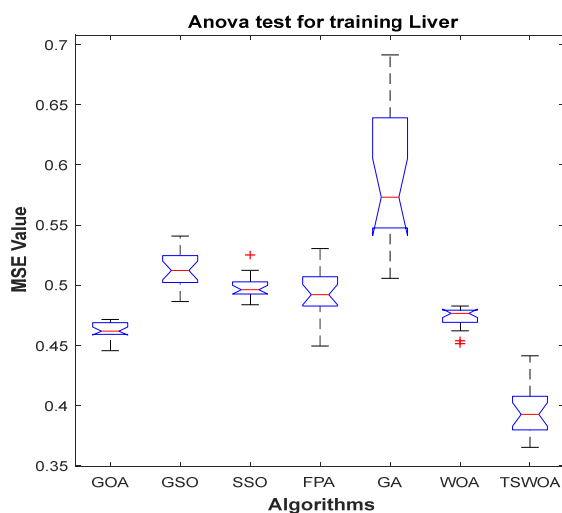
**Figure 9.** The variance diagram of Survival.

#### 6.2.4. Liver classification problem

The experimental results of the Liver classification problem are shown in Table 6, and Figures 10 and 11 display the convergence and variance graphs of all algorithms, severally. As displayed from Table 6, the mean value of MSE obtained by the TSWOA algorithm is slightly smaller than the mean value of MSE obtained by GOA and WOA. In the Liver dataset, the accuracy of the TSWOA algorithm was higher than that of other comparison algorithms, ranking first. It can be distinctly displayed from Figure 10 that at the beginning of the iteration, the TSWOA algorithm not only converges faster but also has a better mean value of MSE than other comparison algorithms. As shown in Table 18, the  $p$  value of the TSWOA algorithm and other algorithms are all less than 0.05, indicating the excellent performance of the modified algorithm in processing the Liver dataset.

**Table 6.** The experimental results of Liver.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	4.46E - 01	4.72E - 01	4.62E - 01	8.53E - 03	60.17	3
GSO	4.86E - 01	5.41E - 01	5.14E - 01	1.54E - 02	55.93	6
SSO	4.84E - 01	5.25E - 01	4.99E - 01	9.82E - 03	56.80	4
FPA	4.49E - 01	5.30E - 01	4.95E - 01	1.82E - 02	62.71	2
GA	5.06E - 01	6.91E - 01	5.93E - 01	5.33E - 02	49.15	7
WOA	4.51E - 01	4.83E - 01	4.73E - 01	8.92E - 03	56.78	5
TSWOA	3.65E - 01	4.41E - 01	3.94E - 01	1.90E - 02	72.88	1

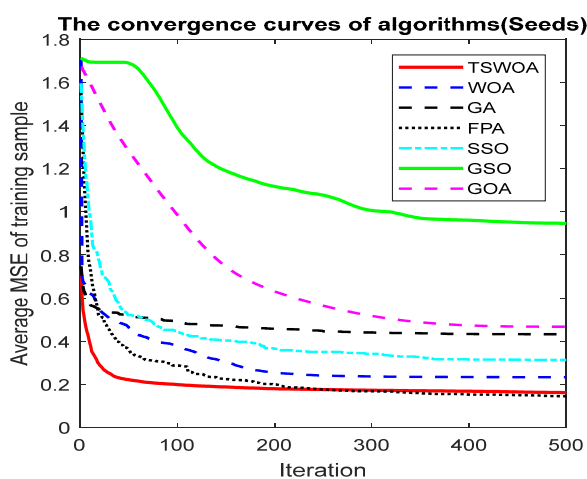
**Figure 10.** The convergent curve of Liver.**Figure 11.** The variance diagram of Liver.

### 6.2.5. Seeds classification problem

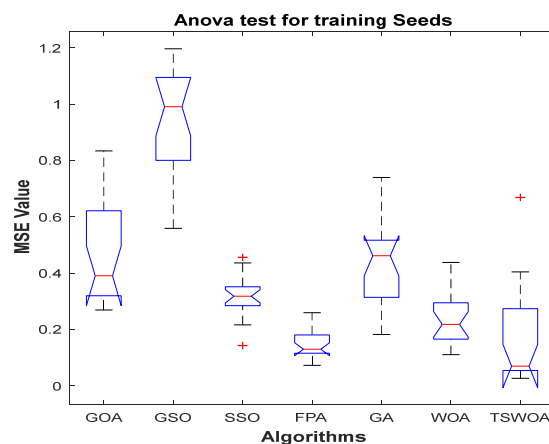
The MSE results obtained by all algorithms listed in Table 7 display that there is little difference between the mean values obtained by the TSWOA algorithm and the FPA algorithm, and the performance of the two algorithms is much better than other algorithms. Meanwhile, the accuracy of both the TSWOA algorithm and the FPA algorithm is 95.77, which is far higher than the accuracy of other comparison algorithms. Figure 12 exhibits that the convergence rate of the TSWOA algorithm is much faster than that of GOA, GSO, SSO, GA, and WOA, but slightly worse than that of FPA. Before 250 iterations, the convergence speed and convergence accuracy of the TSWOA algorithm was much higher than that of other comparison algorithms, but in the later iteration period, the convergence accuracy of the FPA algorithm was slightly higher than that of TSWOA algorithm. Wilcoxon's rank-sum test in Table 18 shows that the value of the TSWOA algorithm and the FPA algorithm are greater than 0.05, but the comparison results of other comparison algorithms are less than 0.05.

**Table 7.** The experimental results of Seeds.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	2.69E - 01	8.34E - 01	4.68E - 01	1.89E - 01	77.46	4
GSO	5.58E - 01	1.20E + 00	9.45E - 01	1.99E - 01	43.66	6
SSO	1.42E - 01	4.56E - 01	3.14E - 01	7.22E - 02	83.10	2
FPA	7.19E - 02	2.59E - 01	1.45E - 01	4.49E - 02	95.77	1
GA	1.82E - 01	7.39E - 01	4.33E - 01	1.45E - 01	70.42	5
WOA	1.10E - 01	4.37E - 01	2.34E - 01	9.62E - 02	81.69	3
TSWOA	2.59E - 02	6.68E - 01	1.62E - 01	1.72E - 01	95.77	1



**Figure 12.** The convergent curve of Seeds.



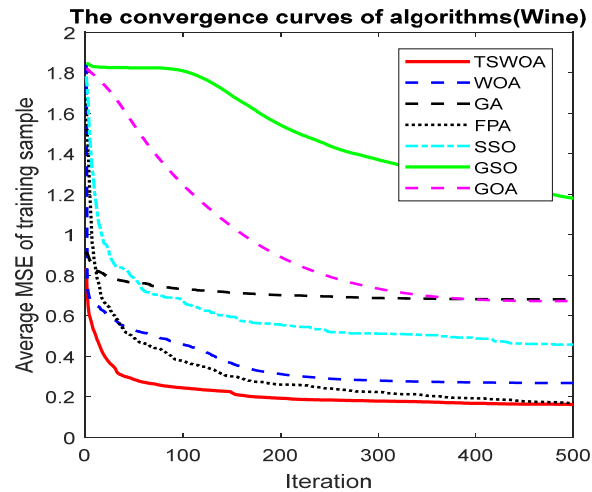
**Figure 13.** The variance diagram of Seeds.

#### 6.2.6. Wine classification problem

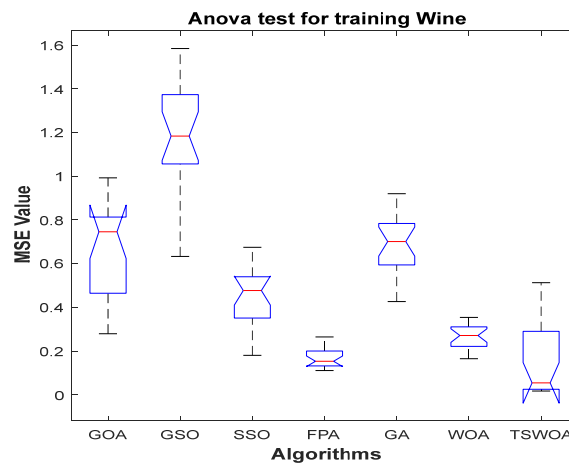
The results in Table 8 display that the TSWOA algorithm obtains better results than the WOA algorithm and the other five algorithms. Table 8 shows that, as far as MSE is concerned, the mean value of TSWOA algorithm is the smallest and the classification accuracy of the algorithm is 96.72, which is much higher than that of other comparison algorithms, indicating that TSWOA algorithm can achieve better classification accuracy than other comparison algorithms. Figure 14 exhibits the convergence curve of the average MSE of all the seven algorithms, which distinctly indicates that GOA, GA, and WOA quickly fall into local optima and their convergence speed has stopped when the number of iterations is 300. Meanwhile, about 500 iterations of FPA and TSWOA reached the corresponding minimum value, but the MSE of TSWOA was smaller than that of FPA. According to the test results in Table 18, the results of the TSWOA algorithm are significantly different from that of GOA, GSO, SSO, GA and WOA algorithms.

**Table 8.** The experimental results of Wine.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	2.79E - 01	9.92E - 01	6.72E - 01	2.18E - 01	67.21	5
GSO	6.34E - 01	1.58E + 00	1.18E + 00	2.46E - 01	44.26	7
SSO	1.81E - 01	6.75E - 01	4.57E - 01	1.35E - 01	81.97	4
FPA	1.11E - 01	2.65E - 01	1.69E - 01	4.59E - 02	85.25	2
GA	4.27E - 01	9.20E - 01	6.81E - 01	1.38E - 01	52.46	6
WOA	1.66E - 01	3.54E - 01	2.68E - 01	5.62E - 02	83.61	3
TSWOA	1.73E - 02	5.13E - 01	1.62E - 01	1.62E - 01	96.72	1



**Figure 14.** The convergent curve of Wine.



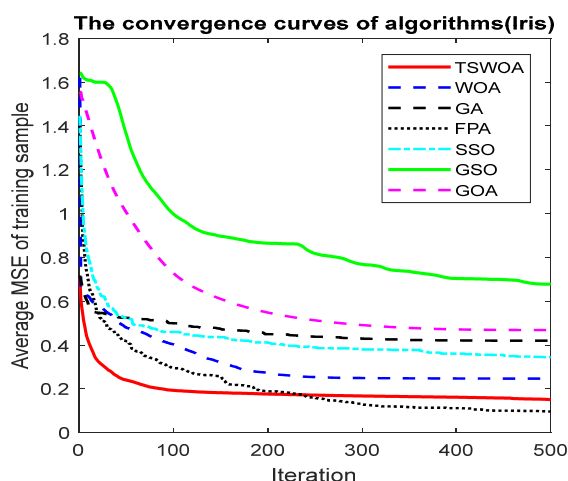
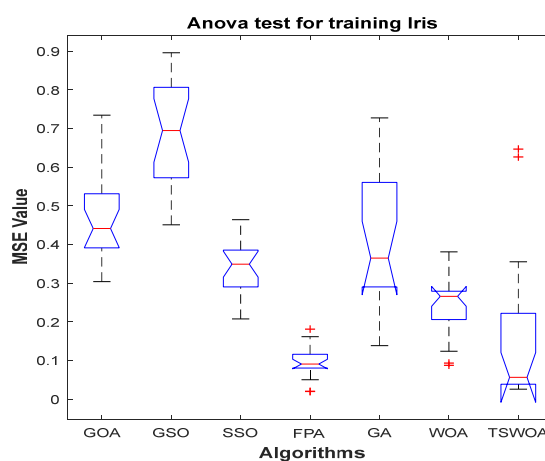
**Figure 15.** The variance diagram of Wine.

### 6.2.7. Iris classification problem

The experimental results of the Iris classification problem are exhibited in Table 9, and Figures 16 and 17 display the convergence and variance graphs of all algorithms, severally. The mean value of MSE obtained by the algorithm shown in Table 9 exhibits that the value obtained by FPA is superior to that of other comparison algorithms, while the value obtained by TSWOA ranks the second. However, the accuracy obtained by TSWOA and GA is much higher than other algorithms. Figure 16 shows that before the 240 iterations, the convergence speed and convergence accuracy of the TSWOA algorithm is much higher than that of other comparison algorithms. But, after the 240 iterations, the convergence accuracy of FPA is greater than that of TSWOA. Meanwhile, GOA, GA, SSO and WOA algorithms quickly fall into local optimization. As displayed in Figure 16, TSWOA has a better variance value than other algorithms. According to the  $p$ -value test in Table 18, the results of TSWOA are obviously differences from those of GOA, GSO, SSO, GA, and WOA.

**Table 9.** The experimental results of Iris.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	3.04E - 01	7.35E - 01	4.68E - 01	1.17E - 01	84.31	6
GSO	4.51E - 01	8.96E - 01	6.77E - 01	1.39E - 01	70.59	7
SSO	2.08E - 01	4.64E - 01	3.44E - 01	6.76E - 02	96.08	3
FPA	2.02E - 02	1.81E - 01	9.64E - 02	4.04E - 02	96.03	4
GA	1.39E - 01	7.28E - 01	4.20E - 01	1.76E - 01	98.03	2
WOA	8.79E - 02	3.81E - 01	2.47E - 01	7.70E - 02	94.12	5
TSWOA	2.63E - 02	6.47E - 01	1.51E - 01	1.90E - 01	98.04	1

**Figure 16.** The convergent curve of Iris.**Figure 17.** The variance diagram of Iris.

### 6.2.8. Statlog classification problem

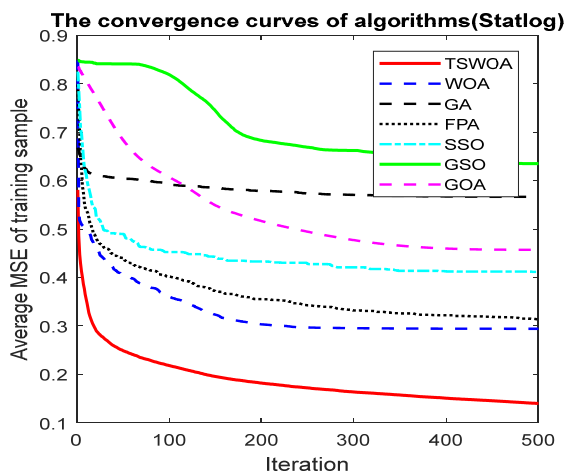
The results in Table 10 exhibit that, in terms of MSE, the mean value and variance of the TSWOA algorithm are the smallest compared with other comparison algorithms. At the same time, the accuracy in Table 10 indicates that the TSWOA algorithm can achieve a better classification effect than other comparison algorithms. Figure 18 displays the convergence curves of MSE obtained by all the seven algorithms, which clearly shows that GSO, GA, SSO, and WOA quickly fall into



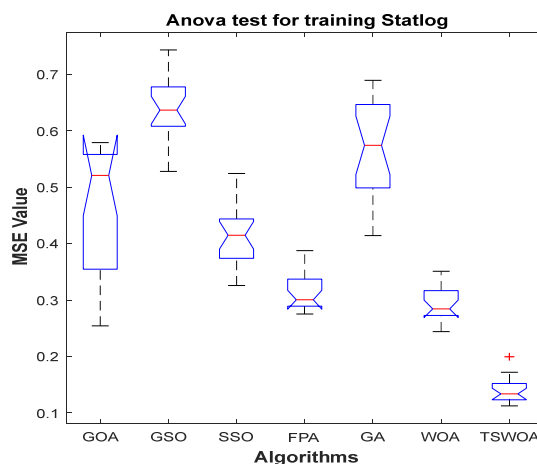
local optima and their convergence speed has stopped when the number of iterations is 200. However, the TSWOA algorithm has been in a declining state in the process of 500 iterations. Figure 19 exhibits that the TSWOA algorithm has a relatively stable variance. Table 18 also displays the superiority of TSWOA, which is obviously different from all algorithms.

**Table 10.** The experimental results of Statlog.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	2.54E - 01	5.79E - 01	4.57E - 01	1.19E - 01	79.35	4
GSO	5.28E - 01	7.44E - 01	6.35E - 01	5.78E - 02	46.74	7
SSO	3.26E - 01	5.25E - 01	4.12E - 01	4.83E - 02	75	5
FPA	2.75E - 01	3.88E - 01	3.13E - 01	3.37E - 02	80.69	3
GA	4.14E - 01	6.90E - 01	5.67E - 01	8.57E - 02	70.65	6
WOA	2.44E - 01	3.51E - 01	2.94E - 01	2.98E - 02	83.69	2
TSWOA	1.12E - 01	2.00E - 01	1.40E - 01	2.23E - 02	83.70	1



**Figure 18.** The convergent curve of Statlog.



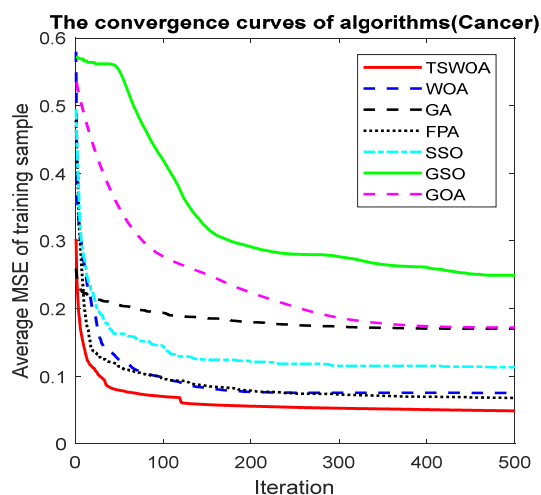
**Figure 19.** The variance diagram of Statlog.

### 6.2.9. Cancer classification problem

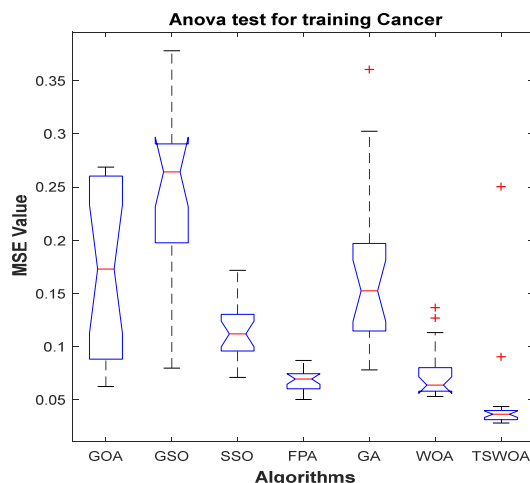
Figure 20 exhibits the convergence of the seven algorithms. As displayed from Figure 20, compared with other comparison algorithms, TSWOA obtained the minimum mean value. FPA is approximately the same mean value obtained by WOA, while the convergence accuracy and speed of the TSWOA algorithm are better than other algorithms. At the same time, it can also be observed that during the optimization process, GA and GOA obtained the same minimum mean, while GSO performed the worst, stagnating in the early stage. Table 11 shows the comparison of the mean values of different algorithms. It can be clearly found from Table 11 that after 500 iterations, the mean value obtained by TSWOA is the smallest. WOA and FPA obtained the same classification effect and ranked first, while TSWOA, GA, and SSO obtained the same classification effect and ranked second. The results above show that TSWOA provides very competitive results in the classification effect and the  $p$ -value in Table 18 further displays that the TSWOA algorithm is significantly superior to all comparison algorithms in the classification effect of the Cancer dataset.

**Table 11.** The experimental results of Cancer.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	6.23E - 02	2.69E - 01	1.72E - 01	7.95E - 02	97	3
GSO	7.96E - 02	3.78E - 01	2.49E - 01	6.78E - 02	89	4
SSO	7.09E - 02	1.72E - 01	1.13E - 01	2.69E - 02	98	2
FPA	5.01E - 02	8.68E - 02	6.79E - 02	9.31E - 03	99	1
GA	7.79E - 02	3.61E - 01	1.70E - 01	7.30E - 02	98	2
WOA	5.29E - 02	1.37E - 01	7.53E - 02	2.57E - 02	99	1
TSWOA	2.78E - 02	2.50E - 01	4.86E - 02	4.93E - 02	98	2



**Figure 20.** The convergent curve of Cancer.



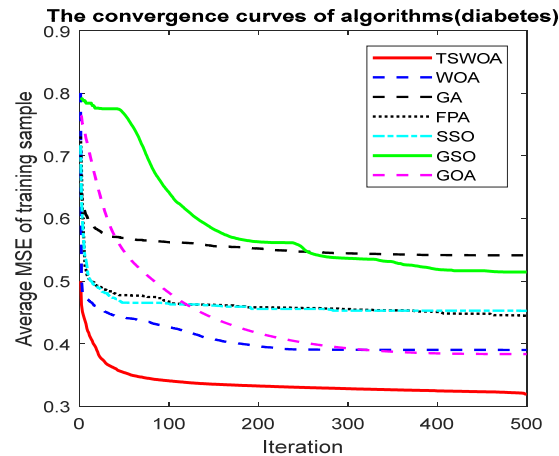
**Figure 21.** The variance diagram of Cancer.

#### 6.2.10. Diabetes classification problem

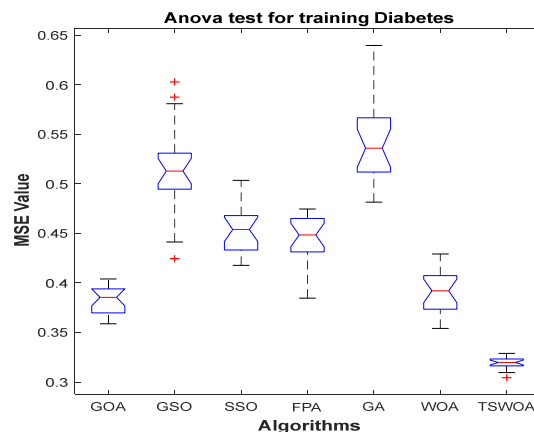
The experimental results of the diabetes classification problem are displayed in Table 12, and Figures 22 and 23 exhibit the convergence and variance of all algorithms, severally. As exhibited from Table 12, as far as MSE is a concern, TSWOA provides the best mean value and the standard deviation of MSE, the result is better than other comparison algorithms, and the accuracy of classification of Diabetes data set is better than that of other comparison algorithms. Figure 22 shows that TSWOA not only has a fast convergence rate but also has a better mean value than other comparison algorithms. In addition, In Figure 23, TSWOA has the smallest variance value. Table 18 also displays the superiority of TSWOA, which is significantly different from all comparison algorithms.

**Table 12.** The experimental results of Diabetes.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	3.59E - 01	4.04E - 01	3.83E - 01	1.41E - 02	75.87	3
GSO	4.24E - 01	6.03E - 01	5.15E - 01	4.54E - 02	64.75	7
SSO	4.18E - 01	5.03E - 01	4.52E - 01	2.35E - 02	65.51	6
FPA	3.85E - 01	4.75E - 01	4.44E - 01	2.48E - 02	80.08	2
GA	4.81E - 01	6.40E - 01	5.41E - 01	4.49E - 02	67.05	5
WOA	3.54E - 01	4.29E - 01	3.90E - 01	2.20E - 02	73.18	4
TSWOA	3.04E - 01	3.29E - 01	3.19E - 01	6.10E - 03	81.23	1



**Figure 22.** The convergent curve of Diabetes.



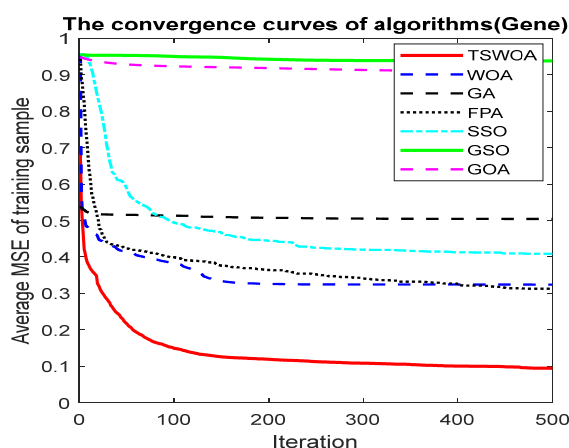
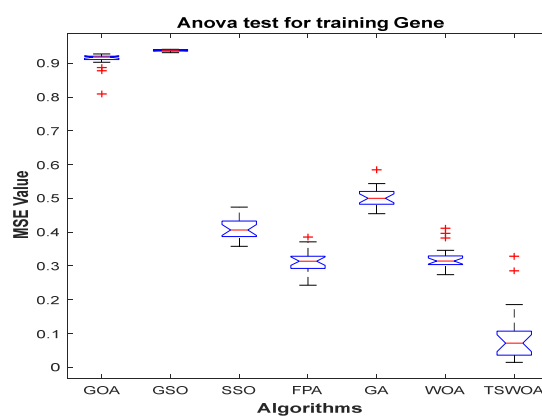
**Figure 23.** The variance diagram of Diabetes.

### 6.2.11. Gene classification problem

The results in Table 13 display that the mean value of MSE obtained by the TSWOA algorithm is better than that of the WOA algorithm and the other five algorithms. The accuracy of the modified algorithm is 33.33, which is much higher than that of other algorithms and ranks first, indicating that the training of the Gene dataset with the TSWOA algorithm can achieve better classification accuracy than other comparison algorithms. Figure 24 exhibits the convergence graphs of all the seven algorithms, which distinctly exhibit that the three algorithms, GSO, GOA, and GA, quickly fall into local optima and their convergence speed has stopped in the early stage of the iteration. However, during the optimization process, TSWOA did not stop still and continued to search for the appropriate minimum. Figure 25 also shows that TSWOA has superior performance. Table 18 displays that TSWOA is obviously different from the other six algorithms. In this case, all values are less than 0.05.

**Table 13.** The experimental results of Diabetes.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	8.09E - 01	9.28E - 01	9.10E - 01	2.68E - 02	2.66	6
GSO	9.31E - 01	9.41E - 01	9.38E - 01	2.67E - 03	1.77	7
SSO	3.58E - 01	4.74E - 01	4.08E - 01	3.07E - 02	5.56	5
FPA	2.43E - 01	3.86E - 01	3.12E - 01	3.80E - 02	13.89	3
GA	4.55E - 01	5.85E - 01	5.04E - 01	3.23E - 02	11.11	4
WOA	2.74E - 01	4.12E - 01	3.24E - 01	3.58E - 02	16.67	2
TSWOA	1.46E - 02	3.29E - 01	9.43E - 02	8.36E - 02	33.33	1

**Figure 24.** The convergent curve of Diabetes.**Figure 25.** The variance diagram of Diabetes.

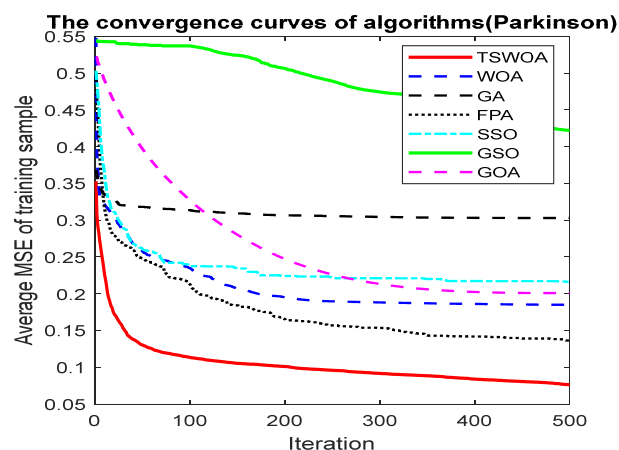
### 6.2.12. Parkinson classification problem

The experimental results of Parkinson's classification problem are shown in Table 14, and the convergence and variance of all algorithms are exhibited in Figures 26 and 27. It displayed from Table 14 that the mean value of MSE obtained by the proposed algorithm is the minimum, indicating that the

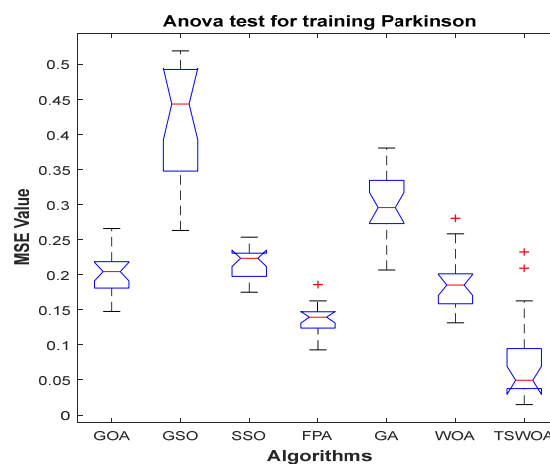
training effect of the improved algorithm is the best. In the test data, the accuracy of the modified algorithm is 74.24, higher than that of other algorithms, ranking first. In Figure 26, the convergence speed of our algorithm is faster than that of other algorithms. It displayed in Figure 27 that TSWOA has a better variance value than other algorithms. Two abnormalities appeared in the experiment, whose average value was less than that of other algorithms. The  $p$ -value test results in Table 18 further indicate that the results of TSWOA are obviously superior to all comparison algorithms.

**Table 14.** The experimental results of Parkinsons.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	1.48E - 01	2.66E - 01	2.01E - 01	3.06E - 02	68.18	5
GSO	2.63E - 01	5.20E - 01	4.22E - 01	8.04E - 02	56.06	7
SSO	1.75E - 01	2.54E - 01	2.16E - 01	2.21E - 02	69.7	4
FPA	9.30E - 02	1.86E - 01	1.36E - 01	1.99E - 02	70.21	3
GA	2.07E - 01	3.81E - 01	3.03E - 01	5.03E - 02	65.15	6
WOA	1.32E - 01	2.81E - 01	1.85E - 01	3.76E - 02	71.21	2
TSWOA	1.48E - 02	2.33E - 01	7.63E - 02	6.08E - 02	74.24	1



**Figure 26.** The convergent curve of Parkinson.



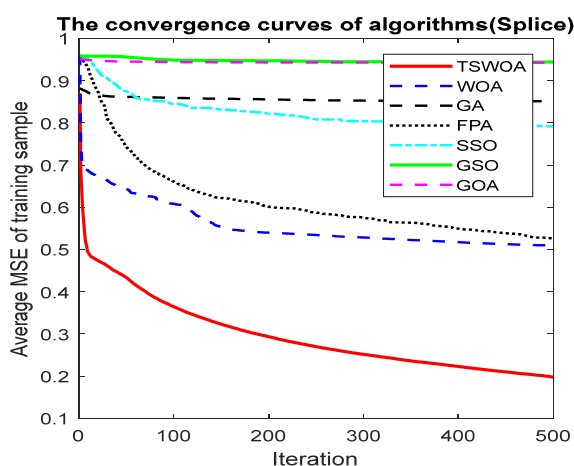
**Figure 27.** The variance diagram of Parkinson.

### 6.2.13 Splice classification problem

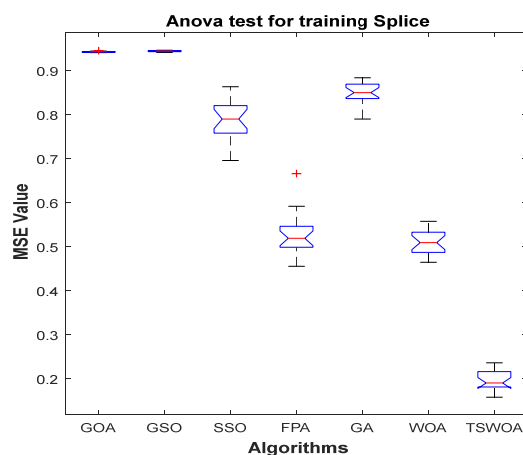
Table 15 displays the experimental results of the splice classification problem, the convergence and variance diagrams of all algorithms are exhibited in Figures 28 and 29, severally. It displayed in Table 15 that the mean value of MSE obtained by the modified algorithm is the minimum. The mean values obtained by GOA and GSO are almost the same. In the Splice dataset, the accuracy of the proposed algorithm is 80, which is higher than the accuracy of other algorithms, so it ranks first. In Figure 28, our algorithm has a great advantage over other algorithms in terms of convergence speed and convergence accuracy. Meanwhile, GSO, GOA, and GA soon fell into local optima. In Figure 29, compared with other algorithms, TSWOA has a better variance value. Experimental results found no singularities, and the mean value of the TSWOA algorithm was lower than that of the other algorithms. In Table 18,  $p$ -value test results display that the TSWOA algorithm is superior to all comparison algorithms.

**Table 15.** The experimental results of Splice.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	9.41E - 01	9.46E - 01	9.43E - 01	1.12E - 03	37.24	7
GSO	9.42E - 01	9.47E - 01	9.45E - 01	1.48E - 03	40.13	6
SSO	6.96E - 01	8.64E - 01	7.92E - 01	4.40E - 02	41.18	4
FPA	4.56E - 01	6.67E - 01	5.27E - 01	4.71E - 02	70	2
GA	7.91E - 01	8.84E - 01	8.52E - 01	2.53E - 02	41.17	5
WOA	4.65E - 01	5.58E - 01	5.10E - 01	2.79E - 02	51.47	3
TSWOA	1.59E - 01	2.37E - 01	1.97E - 01	2.30E - 02	80	1



**Figure 28.** The convergent curve of Splice.



**Figure 29.** The variance diagram of Splice.

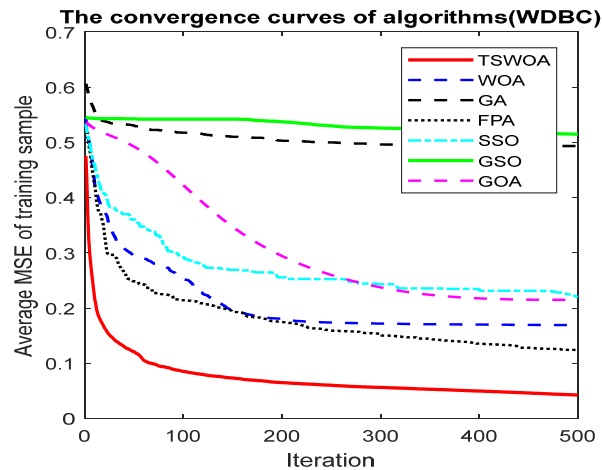
#### 6.2.14. WDBC classification problem

The results in Table 16 show that, as far as MSE is concerned, the mean value and variance of the TSWOA algorithm are the smallest, indicating that the TSWOA algorithm can achieve better classification effect than other comparison algorithms. In the test data, the accuracy of the modified algorithm is 95.15, which is higher than the accuracy of other algorithms, so it ranks first. Figure 30 exhibits that GSO, GA, GOA, and WOA fall into local optimality early in the optimization process, resulting in premature convergence of the algorithm. However, in the optimization process, TSWOA did not stop and continued to search for the corresponding optimal value. All the results exhibit that TSWOA is superior to other algorithms, with the fastest convergence speed and the smallest value. As can be displayed in Figure 31, compared with other algorithms, TSWOA has a better variance value. It can be concluded in Table 18 that TSWOA is obviously different from other comparison algorithms.

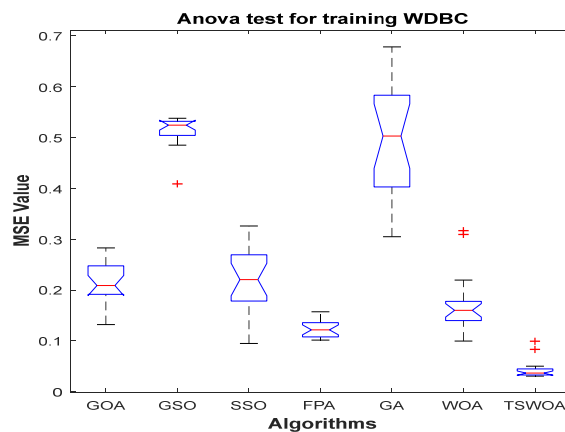
**Table 16.** The experimental results of WDBC.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	1.32E - 01	2.83E - 01	2.15E - 01	4.11E - 02	90.9	5
GSO	4.09E - 01	5.38E - 01	5.15E - 01	2.92E - 02	79.39	6
SSO	9.51E - 02	3.26E - 01	2.20E - 01	5.60E - 02	93.31	4
FPA	1.02E - 01	1.57E - 01	1.24E - 01	1.80E - 02	93.33	3
GA	3.05E - 01	6.79E - 01	4.93E - 01	1.11E - 01	78.79	7
WOA	9.98E - 02	3.17E - 01	1.69E - 01	5.68E - 02	94.55	2
TSWOA	3.07E - 02	9.93E - 02	4.28E - 02	1.77E - 02	95.15	1





**Figure 30.** The convergent curve of WDBC.



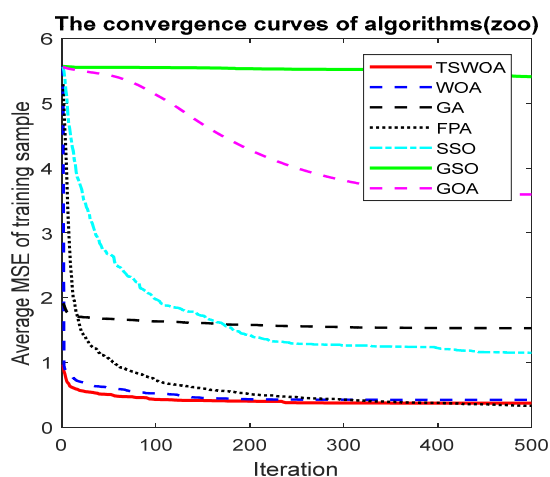
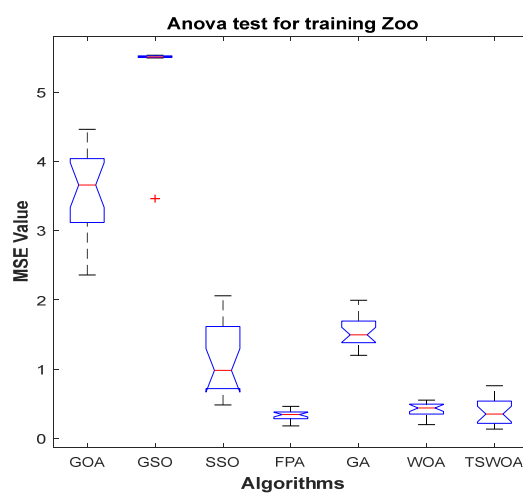
**Figure 31.** The variance diagram of WDBC.

#### 6.2.15. Zoo classification problem

The MSE results obtained by the algorithm displayed in Table 17 exhibit that there is little difference between the mean values obtained by FPA and TSWOA, and the performance of these two algorithms is much better than that of other algorithms. However, the classification accuracy of the modified algorithm is slightly lower than that of WOA in test data. In the Zoo dataset, the accuracy of the modified algorithm is 64.71, which ranks second. The convergence of the seven algorithms is displayed in Figure 32. As can be shown from Figure 32, in the later iteration, TSWOA, WOA and FPA obtained convergence accuracy with little difference. However, GSO and GA tend to fall into local optima in the early stage. As can be seen from Figure 33, compared with other algorithms, TSWOA has a good mean value in training data. The results above exhibit that TSWOA provides very competitive results. It can be concluded from Table 18 that, except for FPA, the values obtained by TSWOA are significantly different from the other five algorithms.

**Table 17.** The experimental results of Zoo.

Algorithms	Best	Worst	Mean	Std	Accuracy	Rank
GOA	2.36E + 00	4.47E + 00	3.60E + 00	5.67E - 01	23.70	7
GSO	3.46E + 00	5.54E + 00	5.41E + 00	4.59E - 01	30.73	5
SSO	4.83E - 01	2.06E + 00	1.15E + 00	5.21E - 01	44.12	4
FPA	1.79E - 01	4.63E - 01	3.32E - 01	7.42E - 02	63.61	3
GA	1.20E + 00	1.99E + 00	1.53E + 00	2.12E - 01	26.47	6
WOA	1.97E - 01	5.52E - 01	4.23E - 01	9.97E - 02	67.65	1
TSWOA	1.34E - 01	7.61E - 01	3.75E - 01	1.85E - 01	64.71	2

**Figure 32.** The convergent curve of Zoo.**Figure 33.** The variance diagram of Zoo.

**Table 18.**  $p$ -values calculated for Wilcoxon's rank-sum test on fifteen data set.

Functions	TSWOA vs					
	GOA	GSO	SSO	FPA	GA	WOA
Blood	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
Scale	6.92E-07	7.90E-08	1.05E-06	1.20E-06	5.23E-07	1.20E-06
Survival	7.90E-08	6.80E-08	6.80E-08	7.90E-08	6.80E-08	7.90E-08
Liver	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
Seeds	4.68E-05	1.23E-07	1.63E-03	6.76E-02	2.60E-05	7.71E-03
Wine	6.01E-07	6.80E-08	1.10E-05	3.30E-01	1.23E-07	2.39E-02
Iris	1.41E-05	5.23E-07	8.29E-05	3.23E-01	3.71E-05	1.01E-03
Statlog	6.80E-08	6.80E-08	6.80E-08	6.73E-08	6.80E-08	6.80E-08
Cancer	1.05E-06	2.22E-07	2.06E-06	1.59E-05	9.13E-07	9.75E-06
Diabetes	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
Gene	6.80E-08	6.80E-08	6.80E-08	9.37E-07	6.80E-08	6.92E-07
Parkinson	5.87E-06	6.80E-08	1.80E-06	2.17E-04	1.06E-07	2.30E-05
Splice	6.80E-08	6.80E-08	6.80E-08	6.79E-08	6.80E-08	6.80E-08
WDBC	6.80E-08	6.80E-08	7.90E-08	6.73E-08	6.80E-08	6.80E-08
Zoo	6.78E-08	6.78E-08	7.93E-07	6.55E-01	6.78E-08	2.61E-01

### 6.3. Result analysis

As can be seen from Tables 3–17, except Survival, Cancer, and Zoo, the accuracy of the modified algorithm is the greatest, indicating that the classification failure rate of TSWOA is small in optimizing MLP. In terms of Survival, Cancer, and Zoo, the accuracy of TSWOA is also in the top three. Moreover, the rest are ranked first. From Figures 4–33, it can be seen that the improved algorithm has faster convergence speed, higher accuracy, and stronger stability for optimizing MLP. The reasons are as follows:

For any optimization algorithm, it is important to obtain the balance between its exploration and exploitation. Exploration makes the algorithm to have feasible solutions and exploitation provides search intensity for any optimization problem. As displayed in section 3. It is vital to have the arguments  $\alpha$  in the position update of WOA. By changing the value of the arguments  $\alpha$ , a better balance can be realized between the global search and local search. However, in the WOA,  $\alpha$  is declining linearly from 2 to 0 with consistently of the iterations. At the beginning of the iteration, the arguments  $\alpha$  can be a max value that is beneficial to global search, but the algorithm has a poor search efficiency. At the end of the iteration, a small value is acquired, which is beneficial to the convergence of the algorithm, but the algorithm is prone to a local optimum. Based on the problems above, the teaching phase of TLBO is add to the basic whale optimization algorithm and selects the best agent to guide other agents to move towards the current optimal position. On the one hand, we heighten the exploitation capacity of the algorithm and improve the quality of candidate solutions. On the other hand, it evades the stagnation state of the algorithm and decreases the risk of the algorithm running into local optimum. In order to strengthen the exploration capacity of the algorithm, the simplex method is added to the algorithm, which not only increases the variety of the population but also enlarges the search scope of the solution space, guiding the algorithm to find the

most promising area of the solution space. Therefore, this modified algorithm can be applied as a new MLP optimizing method.

## 7. Conclusions and future directions

In this paper, an improved teaching learning-based whale optimization algorithm was presented to optimize MLP. The modified TSWOA algorithm utilizes the teacher phase of TLBO to heighten the exploration and increase the overall quality of candidate solutions. The simplex method is added to TSWOA to enlarge the diversification of search agents, which ensures that the algorithm can availablely explore the search location, and make the algorithm quickly converge to the global optimal. This method was adopted to optimize connection weights and biases of multi-layer perceptron (MLP) neural network for the classification problems, compared with several recognized meta-heuristic algorithms. The experimental results display that the modified TSWOA is quite effective in optimizing MLP. Future research may apply TSWOA to optimize convolutional neural networks for image processing. Another opportunity is to further heighten the performance of TSWOA to get a better balance between exploration and exploitation and apply it to solve the real problems.

## Acknowledgments

This work is supported by National Science Foundation of China under Grants No. 61563008, and Project of Guangxi Natural Science Foundation under Grant No. 2018GXNSFAA138146.

## Conflict of interests

The authors declare no conflict of interest.

## References

1. W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, **5** (1943), 115–133.
2. A. Krogh, What are artificial neural networks?, *Nat. Biotechnol.*, **26** (2008), 195–197.
3. A. Leśnia, M. Juszczak, Prediction of site overhead costs with the use of artificial neural network based model, *Arch. Civil. Mech. Eng.*, **18** (2018), 973–982.
4. S. Kaymak, A. Helwan, D. Uzun, Breast cancer image classification using artificial neural networks, *Proc. Comput. Sci.*, **120** (2017), 126–131.
5. J. D. Sitton, Y. Zeinali, B. A. Story, Rapid soil classification using artificial neural networks for use in constructing compressed earth blocks, *Constr. Build. Mater.*, **138** (2017), 214–221.
6. D. Valero, D. B. Bung, Artificial Neural Networks and pattern recognition for air-water flow velocity estimation using a single-tip optical fibre probe, *J. Hydro. Environ. Res.*, **19** (2018), 150–159.
7. C. C. Lai, K. L. Su, Development of an intelligent mobile robot localization system using Kinect RGB-D mapping and neural network, *Comput. Electr. Eng.*, **67** (2018), 620–628.
8. O. Erkaymaz, M. Ozer, M. Perc, Performance of small-world feedforward neural networks for the diagnosis of diabetes, *Appl. Math. Comput.*, **311** (2017), 22–28.

9. N. Wan, W. X. Shi, S. S. Fan, S. X. Liu, PSO-FNNbased vertical handoff decision algorithm in heterogeneous wireless networks, *Proc. Environ. Sci.*, **11** (2011), 55–62.
10. A. A. Hameed, B. Karlik, M. S. Salman, Back-propagation algorithm with variable adaptive momentum, *Knowl. Based Syst.*, **114** (2016), 79–87.
11. D. J. Montana, L. Davis, *Training feedforward neural networks using genetic algorithms*, Proceedings of the 11th International Joint Conference on Artificial Intelligence, 1989. Available from: <https://www.ijcai.org>.
12. C. Reale, K. Gavin, L. Librić, D. Jurić-Kačunić, Automatic classification of fine-grained soils using CPT measurements and Artificial Neural Networks, *Adv. Eng. Inf.*, **36** (2018), 207–215.
13. W. Li, Improving particle swarm optimization based on neighborhood and historical memory for training multi-layer perceptron, *Information*, **9** (2018), 16.
14. S. Gambhir, S. K. Malik, Y. Kumar, PSO-ANN based diagnostic model for the early detection of dengue disease, *New Horiz. Trans. Med.*, **4** (2017), 1–8.
15. S. Z. Mirjalili, S. Saremi, S. M. Mirjalili, Designing evolutionary feedforward neural networks using social spider optimization algorithm, *Neural Comput. Appl.*, **26** (2015), 1919–1928.
16. E. Uzlu, M. Kankal, A. Akpınar, T. Dede, Estimates of energy consumption in Turkey using neural networks with the teaching-learning-based optimization algorithm, *Energy*, **75** (2014), 295–303.
17. I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Comput.*, **22** (2016), 1–15.
18. P. A. Kowalski, S. Łukasik, Training neural networks with Krill Herd algorithm, *Neural Process. Lett.*, **44** (2016), 5–17.
19. D. A. Alboaneen, H. Tianfield, Y. Zhang, *Glowworm swarm optimization for training multi-layer perceptrons*, Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, 2017. Available from: <https://dl.acm.org/doi/abs/10.1145/3148055.3148075>.
20. A. A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, *Soft. Comput.*, **23** (2019), 7941–7958.
21. E. Valian, S. Mohanna, S. Tavakoli, Improved cuckoo search algorithm for feedforward neural network training, *Int. J. Artif. Intell. Appl.*, **2** (2011), 36–43.
22. C. Blum, K. Socha, *Training feed-forward neural networks with ant colony optimization: an application to pattern classification*, Fifth International Conference on Hybrid Intelligent Systems (HIS'05), 2005. Available from: <https://ieeexplore.ieee.org/abstract/document/1587754>.
23. K. Socha, C. Blum, An ant colony optimization algorithm for continuous optimization: Application to feed-forward neural network training, *Neural Comput. Appl.*, **16** (2007), 235–247.
24. D. Karaboga, B. Akay, C. Ozturk, *Artificial Bee Colony(ABC) optimization algorithm for training feed-forward neural networks*, Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence (MDAI '07), 2007. Available from: [https://link.springer.com/chapter/10.1007/978-3-540-73729-2\\_30](https://link.springer.com/chapter/10.1007/978-3-540-73729-2_30).
25. S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptions, *Appl. Intell.*, **43** (2015), 150–161.
26. M. Branch, A multi-layer perceptron neural network trained by invasive weed optimization for potato color image segmentation, *Trends Appl. Sci. Res.*, **7** (2012), 445–455.

27. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82.
28. S. Mirjalili, A. Lewis. The whale optimization algorithm, *Adv. Eng. Soft.*, **95** (2016), 51–67.
29. M. Abdel-Basset, A. N. Hessin, L. Abdel-Fatah, A comprehensive study of cuckoo-inspired algorithms, *Neural Comput. Appl.*, **29** (2018), 345–361.
30. A. A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, *Appl. Soft. Comput.*, **60** (2017), 115–134.
31. S. W. Lin, Z. J. Lee, K. C. Ying, C. Y. Lee, Applying hybrid metaheuristics for capacitated vehicle routing problem, *Expert. Syst. Appl.*, **36** (2009), 1505–1512.
32. J. Chen, B. Xin, Z. Peng, L. Dou, J. Zhang, Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization, *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.*, **39** (2009), 680–691.
33. R. Li, S. Hu, Y. Wang, M. Yin, A local search algorithm with tabu strategy and perturbation mechanism for generalized vertex cover problem, *Neural Comput. Appl.*, **28** (2017), 1775–1785.
34. N. H. Awad, M. Z. Ali, P. N. Suganthan, R. G. Reynolds, CADE: a hybridization of cultural algorithm and differential evolution for numerical optimization, *Inf. Sci.*, **378** (2017), 215–241.
35. M. El-Abd, M. Kamel, A taxonomy of cooperative search algorithms, *Int. Workshop Hybrid Metaheuristics*, **36** (2005), 32–41.
36. V. K. Ojha, A. Abraham, V. Snášel, Metaheuristic design of feedforward neural networks: a review of two decades of research, *Eng. Appl. Artif. Int.*, **60** (2017), 97–116.
37. P. Sarkar, N. M. Laskar, S. Nath, K. L. Baishnab, S. Chanda, Offset voltage minimization based circuit sizing of CMOS operational amplifier using whale optimization algorithm, *J. Inf. Optim. Sci.*, **39** (2017), 1–17.
38. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput. Aided Des.*, **43** (2011), 303–315.
39. W. Spendley, G. R. Hext, F. R. Himsworth, Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, **4** (1962), 441–461.
40. A. A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with le'vy flight for optimization tasks, *Appl. Soft. Comput.*, **60** (2017), 115–134.
41. J. Villanueva, Kolmogorov theorem revisited, *J. Differ. Equations*, **244** (2008), 2251–2276.
42. H. Samet, F. Hashemi, T. Ghanbari, Minimum non detection zone for islanding detection using an optimal Artificial Neural Network algorithm based on PSO, *Renewable Sustainable Energy Rev.*, **52** (2015), 1–18.
43. D. A. Alboaneen, H. Tianfield, Y. Zhang, *Glowworm swarm optimisation for training multi-layer perceptrons*, Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, 2017. Available from: <https://dl.acm.org/doi/abs/10.1145/3148055.3148075>.
44. S. Z. Mirjalili, S. Saremi, S. M. Mirjalili, Designing evolutionary feedforward neural networks using social spider optimization algorithm, *Neural Comput. Appl.*, **26** (2015), 1919–1928.
45. I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Comput.*, **22** (2016), 1–15.
46. H. Wu, Y. Zhou, Q. Luo, M. A. Basset, Training feedforward neural networks using symbiotic organisms search algorithm, *Comput. Intell. Neurosci.*, **2016** (2016).

47. J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.*, **1** (2011), 3–18.
48. J. Li, Q. Luo, L. Liao, Y. Zhou, *Using spotted hyena optimizer for training feedforward neural networks*, International Conference on Intelligent Computing, 2018. Available from: [https://link.springer.com/chapter/10.1007/978-3-319-95957-3\\_88](https://link.springer.com/chapter/10.1007/978-3-319-95957-3_88).



AIMS Press

©2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)