Improved Whale Optimization Algorithm via the Inertia Weight Method Based on the Cosine Function

Xiaoming Shi*, Kun Li, Liwei Jia

Computer Teaching and Research Section Department of Public Infrastructure, Henan Medical College, China
hnsxm1983@yeah.net, sunlik_1982@126.com, zzujialiwei@126.com

Abstract

Whale Optimization Algorithm (WOA) is a new meta-heuristic algorithm proposed by Australian scholar Mirjalili Seyedali in 2016 based on the feeding behavior of whales in the ocean. In response to the disadvantages of this algorithm, such as low solution accuracy, slow convergence speed and easy to fall into local optimum, an improved Whale Optimization Algorithm (IWOA) is proposed in this paper. We introduce chaotic mapping in the initialization of the algorithm to keep the whale population with diversity; introduce adaptive inertia weights in the spiral position update of humpback whales to prevent the algorithm from falling into local optimum; and introduce Levy flight in the random search for food of humpback whales to improve the global search ability of the algorithm. In the simulation experiments, we compare the algorithm of this paper with other metaheuristic algorithms in seven classical benchmark test functions, and the numerical results of four indexes, minimum, maximum, mean and standard deviation, in different dimensions, illustrate that the algorithm of this paper has better performance results.

Keywords: Whale optimization algorithm, Chaos mapping, Inertia weights

1 Introduction

In 2016, Australian scholar Seyedali Mirjalili constructed a new algorithm-Whale Optimization Algorithm (WOA) [1] based on the living predatory behavior of whales in the ocean among natural marine organisms, which has the advantages of simple operation, few parameters, moderate complexity. The algorithm is widely used in engineering field because of its advantages such as simple operation, less parameters, and medium complexity. Like other metaheuristic algorithms, WOA has the disadvantage of fast convergence and easy to fall into local optimum. Therefore, a new solution strategy is proposed in this paper. Strategy 1: To address the problem of lack of diversity in the population, we use chaotic mapping in the initialization phase to maximize the diversity of solutions. Strategy 2: The parameter setting in the spiral position update of WOA introduces an adaptive weight, which makes the individual humpback whales swim in a more reasonable way to avoid the algorithm falling into local optimum by increasing the coefficients of the adaptive weight; Strategy 3: The Levy method is introduced in the humpback whales' search for food behavior to avoid back and forth swimming, thus improving the quality of the global solution of humpback whales. We incorporated these three strategies into WOA to form a new WOA-Improved Whale Optimization Algorithm (IWOA). To verify the performance of IWOA, we select seven classical benchmark functions in our simulation experiments and compare IWOA with classical ant colony algorithm [2], particle swarm algorithm [3] and whale optimization algorithm, and the effect of experimental results shows that IWOA does have better improvement effect.

The structure of this paper is as follows: Section 1 describes the research background., Section 2 describes the current status of WOA research and describes the direction of this paper from these studies. Section 3 shows the process of whales living and feeding in the sea in the form of an algorithm, while Section 4 implements the improvement of WOA from three aspects. Section 5 compares this algorithm with other algorithms in different benchmark functions in simulation experiments to illustrate the effect of the algorithm improvements in this paper, and Section 6 concludes the whole paper.

2 Related Knowledge

To improve the performance of the whale algorithm, scholars have carried out different degrees of research from different aspects. Reference [4] proposed a chaotic strategy-based quadratic opposition-based learning adaptive variable-speed whale optimization algorithm. Simulation experiments show that it can effectively improve the performance of the algorithm; Reference [5] proposed a multistrategy whale optimization algorithm (MSWOA) that is significantly superior and effective in solving global optimization problems; Reference [6] proposed an optimization algorithm called the hunger games search-whale optimization algorithm; Reference [7] proposed the hybrid whale optimization algorithm (HWOA) and combined it with the tabu search algorithm and local search procedures. Reference [8] proposed a strategy for optimizing support vector machines using the whale optimization algorithm, and simulation experiments illustrate that the use of the whale optimization algorithm can improve the prediction performance of support vector machines; Reference [9] proposed a strategy to optimize association rule mining using the whale optimization algorithm, and simulation experiments illustrate that the use of the whale optimization algorithm can improve the

*Corresponding Author: Xiaoming Shi; E-mail: hnsxm1983@yeah.net
DOI: 10.53106/160792642022122307016

effectiveness of association rule mining; Reference [10] proposed improving the WOA by adding Gaussian mutation, chaotic mapping and shrinking strategy methods. The simulation experiment results show that such methods can improve the performance of the WOA and achieve better results in engineering applications; Reference [11] proposed integrating associative learning into the algorithm on the basis of the WOA and obtained a global enhancement algorithm. Simulation experiments showed that it can reduce the running time of the algorithm and increase the speed of operation; Reference [12] proposed adding chaotic mapping to algorithm-LCWOA, and simulation experiments show that the convergence accuracy of the improved algorithm has been significantly improved; Reference [13] proposed the idea of adding local search to WOA, forming a new whale algorithm-LWOA, and simulation experiments of the algorithm show that the convergence accuracy and the running time of the two approaches have different degrees of improvement; Reference [14] proposed a chaotic mapping of WOA, different from the ordinary chaotic mapping, in which the algorithm combines feature selection and chaotic mapping with each other to avoid falling into local search and improves the ability of the algorithm to find the optimal solution; simulation experiments illustrate a significant improvement in performance compared to WOA; Reference [15] proposed an algorithm that integrates the competitive mechanism of multiobjective differential evolution on the basis of the WOA. The algorithm uses the competitive mechanism to select when the humpback whale faces multiple targets, which can effectively improve the performance; Reference [16] proposed to use WOA for SVM optimization, and used it in building geotechnical research and achieved good results; References [17-18] do not propose an application of the whale optimization algorithm, but they provide useful thoughts for this paper.

3 Whale Optimization Algorithm

In nature, whales are a group of animals that obtain food through group behavior. The whale optimization algorithm is a bionic intelligent optimization algorithm that imitates the predation of whales in nature. The predation process of whales is mainly divided into three stages: surrounding predation, bubble attack and hunting for prey.

3.1 Surrounding and Predation

The way whales obtain food in the sea is accomplished by group encirclement. Therefore, at the beginning of the algorithm, due to the lack of prior knowledge, humpback whales first need to determine the approximate location of their prey, and then to obtain the food, the humpback whale calls other whales to the location of the food . In the WOA, because there is no determination of where the food is, it can only be assumed that the current humpback whale's position is the food position (that is, the optimal individual position). Therefore, other whale individuals in the group move toward the current optimal individual position and surround it; then, Formula (1) is used to update the position:

$$X(t+1) = X_p(t) - A\times|C\times X_p(t) - X(t)|. \quad \textbf{(1)}$$

In the formula, $X(t+1)$ represents the position after the $t+1$-th iteration, $X_p(t)$ represents the optimal solution within the current range of the t-th time, and $A\times|C\times X_p(t) - X(t)|$ represents the distance between the current optimal solution and the individual whale. The two very important vectors A and C are expressed as shown in (2) and (3), respectively. $rand_1$ and $rand_2$ denote the random numbers between (0,1) and serve to control the size of the two vectors, while a is the convergence factor with the role of ensuring that the two vectors A and C have a certain convergence so that the algorithm avoids failure to converge, and the value of the decreasing trend in is set in [2,0]. $t_{\max}$ denotes the maximum number of iterations

$$A = 2a\times rand_1 - a. \quad \textbf{(2)}$$

$$C = 2\times rand_2. \quad \textbf{(3)}$$

$$a = 2 - 2t/t_{\max}. \quad \textbf{(4)}$$

In the formula, $t_{\max}$ is the maximum number of iterations.

3.2 Spiral Bubble Attack

After the whale obtains the position of the food, it does not directly take the food since unique air bubbles from the whale's head are first used to attack the food, and then it knocks the food out and obtains the food. The WOA simulates the whale through contraction and envelopment behavior and spiral renewal behavior. To prey on the behavior of spitting out bubbles, the goal of the WOA algorithm is set to obtain the local optimal solution.

(1) Shrink enveloping mechanism

In Formula (1), the individual whale approaches the optimal solution position. In the formula, the factor $|A|<1$ plays a more critical role. From the formula, when $|A|<1$, the individual whale is approaching the whale in the current optimal position, and $|A|$ is the size of the array that determines the size of the whale's moving pace.

(2) Spiral update position

Before obtaining food, the whale needs to calculate the distance between the individual whale and the food. The whale does not blindly call other whales to approach the food immediately since it needs to estimate its own position and the position between where the food is located The whale does not directly rush to the food but takes a spiral approach to search and locate the prey. Therefore, in the algorithm, the spiral update is expressed in Equation (5):

$$X(t+1) = D^{'}\times e^{lb}\times\cos(2\pi l) + X_p(t). \quad \textbf{(5)}$$

In the formula, $D^{'} = |X_p(t) - X(t)|$ is used to represent the distance between the i-th whale and its prey, the parameter b is mainly used for the shape constant when the whale is moving in a spiral, and l represents a random number between -1 and 1. The cosine function can be used to express the state when the position is updated, and the probability p represents the choice of the balance enveloping mechanism and the spiral position. According to the algorithm requirements, the value is set as 0.5.

3.3 Random Search for Prey

Individual whales can randomly swim in all directions to find food. Of course, this behavior is a random process. The essence of searching is also to determine a new location based on the location of other whales, expressed as follows:

$$X(t+1) = X_{rand}(t) - A|C \times X_{rand}(t) - X(t)|. \tag{6}$$

In the formula, $X_{rand}(t)$ is the position of the individual whale randomly selected in the current population.

4 Improved Whale Optimization Algorithm

This paper proposes the improved whale optimization algorithm (IWOA), which is improved from the following three aspects.

4.1 Population Initialization

There is no description of the initialization of the population in WOA, which is the reason why the algorithm is prone to fall into local optimum. We initialize the population of WOA using chaotic mapping. Since chaotic mapping has good randomness, ergodicity and periodicity, it can be processed for individual solutions in WOA, which maintains the diversity of individual solutions of abundant fish and thus improves the quality of individual solutions.

$$x_{k,j} = x_{k-1,j} + 0.1 \times rand(0,1). \tag{7}$$

$$x_{i,j} = x_{\min,j} + x_{i,j} \times (1 - x_{\min,j} / x_{\max,j}). \tag{8}$$

In Formulas (7-8), $k-1$ and k represent the number of iterations, j represents the dimension, and i represents the individual. In particular, in Equation (8), the individual $x_{i,j}$ in each iteration needs to be mixed with the maximum value $x_{\max,j}$ and minimum value $x_{\min,j}$ obtained by the individual at the current number of iterations of that individual, and to be able to guarantee the effect of chaotic mapping, it is also necessary to perform the cumulative operation on the minimum value of the individual to ensure the effect of individual chaos. Through such a chaotic operation, the individual has more solution diversity in the population, which provides a better guarantee for the subsequent generation of optimal solutions.

4.2 Inertia Weights based on the Cosine Function

Most metaheuristic algorithms fall into a local optimum when solving the optimal solution [19]. Generally, we use a linear weighting method in the update of the individual solution, but this method can only guarantee that it will not fall into a local optimum within a certain number of iterations, while the linear weights lack variation, so the individual solution at a later stage may also fall into a local optimum. In order to avoid this situation, we use the adaptive weighting method based on the cosine function. The purpose of using this method is to fully consider the iterative characteristics of the whale optimization algorithm, through the periodic characteristics of the cosine function, so that the individual is guaranteed to avoid falling into a local optimum.

$$\lambda = \cos\left(\frac{2\pi t}{t_{\max}}\right) + \frac{2}{f_{obj}^{\max}(x_i^t) + f_{obj}^{\min}(x_i^t)}. \tag{9}$$

In the formula, $t_{\max}$ is the maximum number of iterations, t is the current number of iterations, the fitness value of the individual is added to better integrate the individual and the weight so that the individual has optimizing ability, $f_{obj}^{\max}(x_i^t)$ and $f_{obj}^{\min}(x_i)$ represent the maximum and minimum individual fitness of the current individual i during the t iterations, respectively. From the perspective of the overall execution of the algorithm, at the beginning of the algorithm, the value of λ is relatively large, so the algorithm mainly performs a global search. As the number of iterations continues to increase, the value of λ gradually decreases. In particular, the appearance of the cosine function is able to maintain the vitality of the individual's solution on the one hand, and on the other hand, it has a good effect on the convergence accuracy of the algorithm and the exact accuracy of the solution and avoids the possibility of the individual falling into local search prematurely. Therefore, in view of the above analysis, the spiral update Formula (5) in the WOA is modified in this paper, and the formula is as follows.

$$\overrightarrow{X}(t+1) = \overrightarrow{D'} \times e^{lb} \times \lambda + \overrightarrow{X*}(t). \tag{10}$$

4.3 Optimization of Individual Foraging Behavior by Levy Flights

In the study of bionic animals, the scholar EDWARDS AM [20] found that bionic animals randomly advance in any dimensional space with an arbitrary length of distance in any direction. This behavioral characteristic is called the Levy flight characteristic. On the one hand, this kind of flight feature can perform a local search in a small area, and on the other hand, it can perform a global search in a large area. The operation of a Levy flight can help individuals in the WOA achieve a certain balance between the local scope and the global scope, thereby improving the quality of the global optimal solution. Therefore, in the WOA description in this paper, the original Levy flight characteristics are specifically optimized.

$$Levy(s):| s |^{-1-\beta}, 0 < \beta \leq 2. \tag{11}$$

In Formula (11), s is a random step size, and a Levy flight refers to this random step size. Therefore, s is expressed as follows

$$s = \mu / | v |^{1/\beta} . \tag{12}$$

The expression of s is specific to Levy flight behavior in a way that its value relies entirely on the expression of the parameters, and in regard to the two parameters in Eq(12). μ, v , which fully comply with the normal distribution, each has the following expressions.

$$\mu: N(0, \sigma_\mu^2), v: N(0, \sigma_v^2). \tag{13}$$

In the formula

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1$$

From the above behavior of Levy flight characteristics, whales also have such behaviors in the predation process, especially in the foraging stage. Humpback whales lack prior knowledge and guide other whales to approach it, which easily causes the algorithm to fall into a local optimal situation. Therefore, to avoid this situation, it is necessary to introduce the Levy flight mechanism into the foraging behavior of the algorithm. Therefore, for this reason, this article will update the foraging behavior after the introduction of the Levy flight mechanism as follows:

$$X(t+1) = X(t) + a(t) \times sign(rand) \oplus s . \tag{14}$$

In Eq. (14), $rand$ replaces the original p , and we believe that choosing a random number between [-1,1] is more suitable for the algorithm in this paper. We use $sign(rand)$ to denote the Levy flight characteristic function with parameter $rand$, as shown in Equation (15), and in addition, we choose $a(t)$ as the scale factor function for the number of iterations, this ensures that the algorithm can obtain better results in finding the optimal solution. as shown in Equation (16).

$$sign(rand) = \begin{cases} 1 & rand \in [0,1] \\ -1 & rand \in [-1,0) \end{cases} . \tag{15}$$

$$a(t) = \exp(-t / f_{obj}^{best}(x_i^t) - f_{obj}(x_i^t)) . \tag{16}$$

In Formula (15), when the value of $rand$ is greater than or equal to 0, the Levy flight characteristic function takes the value 1; otherwise, the value is -1. In Eq. (16), $f_{obj}(x_i^t)$ and $f_{obj}^{best}(x_i^t)$ represent the fitness value of the i th individual whale at the t th iteration and the population optimal individual fitness value of the current population, respectively. Formula (14) shows that the update of foraging behavior is significantly improved compared to Formula (6). After using the Levy flight feature, the WOA searches for the optimal solution in a small range in the initial stage of the algorithm. As the number of iterations continues to increase, a random search is performed in a large range, which can ensure the search effect of the WOA in different ranges. It is possible to obtain more high-quality optimal solutions. From the above research results, it can be found that the WOA uses the Levy flight characteristic function to solve the problem of oscillation near the extreme value in the process of generating the optimal solution of the algorithm, thereby improving the efficiency of the algorithm to obtain the optimal solution.

4.4 Algorithm Complexity Analysis

Algorithm complexity is an important reference for testing the performance of an algorithm. The complexity is divided into time complexity and space complexity. The space complexity of this article has not changed, and the time complexity is mainly considered, so this article chooses time complexity to measure the performance of the algorithm. The so-called time complexity mainly measures the workload required during the execution of the algorithm. The time complexity of the WOA algorithm comes from the influence of many factors, such as the population size N , the search dimension D , and the number of iterations T . Therefore, the overall time complexity of the WOA algorithm is d, while the time complexity of the IWOA algorithm proposed in this paper has a significant improvement compared to the time complexity of the WOA, which is reflected in the increase of $O(T)$ for population initialization, $O(T)$ for adaptive weight, and $O(N \times T)$ for Levy flight behavior.

4.5 Algorithm Flow

Step 1: Initialize the parameters of WOA and set the maximum number of iterations
Step 2: In the individual initialization, use Eqs. (7-8) to complete the initialization process for all individuals.
Step 3: In whale individual spiral, use Eq. (10) to perform individual update of individual position.
Step 4: In the whale search for food, use formula (14) to complete the individual search
Step 5: Increase the number of iterations by 1
Step 6: When the number of iterations reaches the maximum, the algorithm ends, otherwise go to step 3.

5 Simulation Experiment

5.1 Algorithm Settings

To better verify the performance improvement effect of the IWOA algorithm, this paper chooses the ACO, PSO, WOA, CWOA [12], and LWOA [13] algorithms to compare with the algorithm in this paper. The computer hardware platform was a Core i7 processor with 16 GHz memory and a 1000 G hard disk. The operating system was Win10, and the simulation software was MATLAB2012a. The population size is set to 100, and the number of iterations is set to 100. The parameters required by the various algorithms are shown in Table 1.

Table 1. The main parameters of the 6 algorithms

Algorithm	Parameter description
ACO	The pheromone value is set to 0.01, the volatilization coefficient is set to 0.01, and the path selection probability is set to 0.5.
PSO	The inertia weight is set to 0.1, and the learning factor is 0.5.
WOA	a is [2,0] linearly decreasing
CWOA	a is [2,0] linearly decreasing, and the chaotic mapping value is 0.5.
LWOA	a is [2,0] linearly decreasing, and the value of β is 1.5.
IWOA	a is [2,0] linearly decreasing, the value of β is 1.5, and the value of *rand* is 1.

5.2 Classic Test Function

This paper selects 7 representative test functions (Table 2) to evaluate the performance of the algorithm in this paper. The reason for choosing these classic test functions is that these functions can measure whether the algorithm in the high and low dimensions of this paper can converge or reach the accuracy that the algorithm can achieve. Such a comparison can theoretically illustrate the performance advantage between the algorithm in this paper and the comparison algorithm. The choice of test metrics is a matter of core illustrative power of the algorithm results, and we have chosen minimum, maximum, mean, and standard deviation as the metrics. These metrics have always been important indicators for algorithm performance measurement. Among these four metrics, the first two metrics mainly measure the quality effect of the solution, the third metric is used to measure the accuracy of the solution as required, and the fourth metric compares the effect of the solution with different numbers of iterations in different dimensions.

Table 2. Test function

Number	Function name	Test function expressions
F1	Sphere	$f(x)=\sum_{i=1}^{n} x_i^2$
F2	Schwefel2.22	$f(x)=\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
F3	Schwefel1.2	$f(x)=\sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)$
F4	Schewfel2.21	$f(x)=\max(abs(x_i))$
F5	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
F6	Step	$f(x)=\sum_{i=1}^{n}([x_i+0.5])^2$
F7	Rastrigin	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$

5.3 Analysis of Experimental Results

Table 3 to Table 9 shows the comparison results of the algorithm performance metrics of the six algorithms in the seven classical test functions. Next, we analyze the test results of these seven classical functions separately.

F1 benchmark function: this algorithm has obvious advantages over the other five algorithms in four indicators, especially in the higher dimension (such as dimension 30), the value of the algorithm’s indicators are the smallest, especially when the dimension is 2, the indicator minimum results equal to 0, which indicates that the performance of this algorithm is very good, also in the dimension of 5 and 10, this algorithm performance is as good.

F2 benchmark function: the algorithm in this paper is smaller in dimension (such as dimension 2) when the values are the smallest, which shows that the quality of the algorithm’s solution has good stability, in dimension 5 and 10, the performance of this algorithm exceeds that of other comparative algorithms.

F3 benchmark function: the algorithm of this paper maintains the best values regardless of whether the dimensional values are small or large, which shows that the algorithm has better performance in four different dimensional values and also shows that the performance of the algorithm is better.

F4 benchmark function: this algorithm has better performance no matter when the dimensional value is small or large, especially when the minimum value indicator is once again 0 in dimension 2, and when other dimensions (such as 5, 10, 30), the four indicators of this algorithm is only higher than the comparative CWOA, the advantage is not obvious, but compared to ACO, PSO and WOA, this algorithm indicator data results are satisfactory.

F5 benchmark function: the algorithm of this paper, whether in the smaller or larger dimensional values, the algorithm of the values remain optimal, although there is no minimum value of 0, but the performance of the algorithm still withstood the test, here it should be noted that when the dimension of 30, this algorithm corresponds to the maximum and average of the two indicators, this algorithm has only a slight advantage over CWOA.

F6 benchmark function: the algorithm in this paper, whether the value of the dimension is small or large, the algorithm’s values remain optimal, although there is no minimum value of 0, but the performance of the algorithm still withstood the test, especially when the dimension is 10, the algorithm compared to LWOA, WOA algorithm has the advantage is not very obvious, and in the dimension of 2, the algorithm’s data results have obvious advantages

F7 benchmark function: the algorithm of this paper, whether in the smaller or larger dimensional values, the algorithm of the values remain optimal, although there is no minimum value of 0, but the performance of the algorithm still performs well, especially in various dimensions have better results. And when other dimensions (such as 5,10,30), the four indicators of this algorithm are only higher than the comparative CWOA, although the advantage is not obvious, but compared to ACO, PSO and WOA, this algorithm index data results are satisfactory.

From the above results, it is found that the overall performance of the algorithm in this paper has improved significantly compared to WOA. In the analysis of the complexity of the algorithm, although the complexity of the algorithm in this paper has improved, the performance of the algorithm is still good, and the algorithm in this paper has better results in all four data indicators. Although the advantage is not obvious in the individual index data, it does not prevent the effect of the improvement of the algorithm in this paper. Therefore, in summary, the algorithm performance of this paper still has some advantages.

Table 3. Comparison results of F1 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.0203476	16456.500	1989.67792	4983.5158
	5	0.3707979	30581.9835	4471.44257	8648.6628
	10	562.4510	57141.6827	16120.5908	17758.7147
	30	26114.2603	132024.456	69169.7457	35162.8114
PSO	2	6.3461e-10	0.0021637	0.000139559	0.0003861
	5	0.005738	4.98512	1.01386505	1.3156749
	10	2.29394	481.84602	145.480588	105.66809
	30	1442.3533	6300.0227	3537.511730	1210.68219
WOA	2	1.441E-14	2.814E-05	1.254E-06	4.485E-06
	5	2.059E-07	5.326E-02	5.942E-03	1.082E-02
	10	1.168E-04	1.098E+00	9.542E-02	1.679E-01
	30	2.649E-03	1.041E+01	1.702E+00	2.222E+00
CWOA	2	9.0480E-18	2.6337E-07	1.1068E-08	3.9676E-08
	5	2.4659E-18	3.3345E-04	1.5645E-05	6.2494E-05
	10	4.9822E-16	4.4317E-04	1.8426E-05	6.9441E-05
	30	4.2062E-13	5.2668E-03	1.2560E-04	7.4489E-04
LWOA	2	6.9109E-16	2.1953E-04	4.6354E-06	3.1022E-05
	5	6.5354E-08	1.1884E-01	9.3972E-03	2.2033E-02
	10	8.4292E-06	6.7253E-01	9.2900E-02	1.2863E-01
	30	4.5597E-03	6.7957E+00	1.5615E+00	1.7818E+00
IWOA	2	0	1.1738E-67	2.4116E-69	1.6594E-68
	5	3.8768E-55	8.1156E-41	1.8560E-42	1.1518E-41
	10	2.0110E-49	6.0219E-35	1.3795E-36	8.5234E-36
	30	5.6653E-43	1.7244E-31	4.8737E-33	2.5423E-32

Table 4. Comparison results of F2 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.171701	90.7335	9.18269	24.75729
	5	0.847493	16850.7147	798.050	2890.376
	10	4.727616	333392446.90	10854563	49590932.044
	30	44551072.64	1.66977e+20	3.41849e+18	2.36061e+19
PSO	2	4.61154e-06	0.00995994	0.00100221	0.00200884
	5	0.004236052	1.2703288	0.2266671	0.2595154
	10	1.0877692	8.44706953	4.5504297	1.89244510
	30	15.01764810	66.971699	32.4898099	10.564354
WOA	2	1.921E-07	5.808E-03	3.905E-04	9.117E-04
	5	1.316E-04	1.798E-01	3.832E-02	3.802E-02
	10	2.252E-03	1.471E+00	3.436E-01	3.124E-01
	30	1.218E-01	8.858E+00	2.126E+00	2.089E+00
CWOA	2	4.2539E-09	4.4962E-04	5.8668E-05	1.0604E-04
	5	3.5113E-09	2.5201E-02	1.7769E-03	4.4900E-03
	10	7.0585E-08	4.8661E-02	3.5748E-03	9.3047E-03
	30	1.4331E-06	7.0166E-01	3.2229E-02	1.1160E-01
LWOA	2	2.0348E-06	5.3100E-03	3.5035E-04	8.7636E-04
	5	7.6096E-04	2.7352E-01	6.8806E-02	6.4547E-02
	10	3.5134E-02	1.3472E+00	3.3425E-01	3.0949E-01
	30	1.3783E-01	8.9488E+00	2.4873E+00	1.9442E+00
IWOA	2	0	5.383E-42	3.172E-43	1.052E-42
	5	1.277E-32	6.883E-26	2.344E-27	1.025E-26
	10	8.473E-30	1.525E-22	5.799E-24	2.216E-23
	30	3.240E-28	2.511E-21	2.215E-22	5.730E-22

Table 5. Comparison results of F3 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.025686	270.41155	38.91627	88.25971
	5	0.018994	1017.43652	96.19119	273.9338
	10	0.176349	3131.65679	280.8745	828.66022
	30	0.159286	13060.0157	758.43687	2825.95712
PSO	2	4.0795811e-11	0.0016373	9.93502e-05	0.00028406
	5	9.6079588e-11	0.00144573	0.0001186	0.00025395
	10	1.91657e-09	0.01397033	0.0005589	0.00206236
	30	8.92441e-11	0.039662004	0.0017298	0.0059671
WOA	2	6.880E-11	2.115E-04	2.178E-05	3.788E-05
	5	8.461E-07	3.365E-03	2.661E-04	5.333E-04
	10	4.719E-07	1.899E-02	1.661E-03	3.048E-03
	30	3.727E-05	9.489E-02	1.306E-02	1.983E-02
CWOA	2	1.4350E-09	8.5384E-03	3.7521E-04	1.3676E-03
	5	2.7207E-10	1.5871E-01	1.3300E-02	2.9698E-02
	10	3.8822E-07	7.2073E-01	4.9116E-02	1.3323E-01
	30	4.6607E-06	9.0665E+00	1.0012E+00	1.9562E+00
LWOA	2	5.4324E-08	1.8933E-04	1.9570E-05	3.7297E-05
	5	5.4736E-07	3.0950E-03	3.2137E-04	5.2946E-04
	10	1.6229E-06	1.3668E-02	1.7022E-03	3.2095E-03
	30	8.8188E-05	3.1915E-01	1.7030E-02	4.5900E-02
IWOA	2	0	3.7874E-10	3.6633E-11	7.0173E-11
	5	1.2057E-12	5.9425E-09	5.9048E-10	1.1155E-09
	10	1.1827E-11	1.6488E-08	2.5080E-09	4.0931E-09
	30	4.5880E-11	2.8673E-07	2.3932E-08	5.0602E-08

Table 6. Comparison results of F4 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.15161	98.9014	40.89212	44.35097
	5	3.97965	98.8703	77.03869	28.64099
	10	28.8742	99.63691	90.60360	12.575049
	30	87.06238	99.83433	96.58320	2.8063077
PSO	2	2.3331863e-05	0.1167497	0.0079861	0.01785393
	5	0.0515829	4.4855845	0.8557691	0.88475519
	10	4.1968123	19.512747	10.652086	3.66702843
	30	15.841233	39.906120	26.712640	4.98594719
WOA	2	9.173E-07	2.349E-01	1.893E-02	4.220E-02
	5	2.922E-02	7.346E-01	2.728E-01	1.487E-01
	10	1.823E-01	9.975E-01	6.390E-01	2.171E-01
	30	6.550E-01	1.697E+00	1.333E+00	2.146E-01
CWOA	2	7.4083E-10	4.8827E-04	3.0697E-05	8.2312E-05
	5	1.4428E-10	3.8148E-03	1.8466E-04	6.8415E-04
	10	3.3838E-09	1.6005E-02	4.9804E-04	2.3528E-03
	30	2.1878E-10	1.3632E-02	5.1994E-04	2.1744E-03
LWOA	2	4.9197E-06	1.5558E-01	1.2626E-02	2.4664E-02
	5	1.1256E-02	7.4553E-01	2.7714E-01	1.6328E-01
	10	3.5382E-01	1.0113E+00	6.9495E-01	1.6096E-01
	30	1.0367E+00	1.7317E+00	1.3670E+00	1.6881E-01
IWOA	2	0	3.5239E-15	1.4043E-16	5.8358E-16
	5	2.1804E-13	3.0607E-02	1.2986E-03	4.9925E-03
	10	7.3759E-06	4.8473E-01	7.7337E-02	1.0178E-01
	30	3.8849E-02	6.1964E-01	4.3101E-01	1.5206E-01

Table 7. Comparison results of F5 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.1722043	79962962.92	7491663.86	22733221.52
	5	18.845310	167820742.75	23591724.06	55016275.21
	10	259.87970	323733723.58	45288706.35	99657763.65
	30	50005.033	776897069.35	144735089.29	274523195.49
PSO	2	4.536701e-07	7.70186	0.2493286	1.10263
	5	1.6381553	1731.353	165.48465	360.10007
	10	216.66020	67022.206	8255.2939	13456.5928
	30	135286.400	2905425.250	847467.368	634605.300
WOA	2	7.135E-09	2.314E+01	7.926E-02	2.812E-01
	5	8.137E-02	6.919E+02	3.912E+01	2.349E+00
	10	7.374E+01	1.716E+02	2.338E+01	3.318E+01
	30	2.312E+02	8.916E+03	2.816E+02	2.713E+02
CWOA	2	0.6015	0.9892	0.6557	0.0939
	5	3.5985	3.9743	3.8919	0.1064
	10	8.4986	9.0698	8.8758	0.0954
	30	28.4938	28.8485	28.7042	0.0466
LWOA	2	0.00003	0.17274	0.02130	0.03667
	5	1.00388	8.45112	3.92143	1.48542
	10	8.94112	187.97241	27.83011	35.16632
	30	29.39202	1274.87336	363.91765	310.45510
IWOA	2	7.1962E-11	4.5806E-04	1.5721E-05	6.5310E-05
	5	1.6594E-03	3.0692E+00	9.9933E-01	8.4164E-01
	10	3.1007E-01	8.9153E+00	6.9537E+00	1.6665E+00
	30	2.7516E+01	2.8816E+01	2.8286E+01	4.0360E-01

Table 8. Comparison results of F6 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	0.03378920	16638.3892	1698.4606	4631.1454
	5	0.31684722	30570.9268	4916.7122	9140.7732
	10	611.799391	56516.6916	16125.4211	17752.7511
	30	26202.1263	132628.8162	69217.9809	35142.1199
PSO	2	2.242653e-10	0.00459714	0.00033719	0.00085328
	5	0.0004307558	12.333401	1.12429536	2.12086752
	10	12.1117574	560.90399	143.552158	115.685413
	30	1363.98861	7473.0983	3643.11901	1311.62557
WOA	2	5.704E-10	9.926E-04	6.431E-05	1.951E-04
	5	8.987E-04	2.929E-01	4.114E-02	4.675E-02
	10	9.370E-02	1.588E+00	5.122E-01	2.791E-01
	30	1.314E+00	1.214E+01	4.299E+00	2.124E+00
CWOA	2	8.0095E-09	2.3722E-02	1.5786E-03	4.2013E-03
	5	5.9473E-11	4.3627E-01	1.5907E-02	6.3788E-02
	10	6.4474E-06	1.7708E+00	1.1807E-01	2.8349E-01
	30	1.8403E-06	3.9997E+00	3.6216E-01	7.8134E-01
LWOA	2	3.7402E-08	6.6989E-04	3.0574E-05	9.9152E-05
	5	3.2406E-04	2.9195E-01	4.7375E-02	6.4563E-02
	10	1.3796E-01	1.4132E+00	5.5585E-01	2.9196E-01
	30	1.3178E+00	1.2369E+01	4.0855E+00	2.0166E+00
IWOA	2	1.4320E-14	6.0278E-08	4.7190E-09	1.2007E-08
	5	1.2199E-07	2.4806E-01	9.6201E-03	4.7534E-02
	10	2.2489E-04	3.0199E-01	8.5553E-02	1.1095E-01
	30	3.7865E-01	2.5983E+00	1.4056E+00	6.0852E-01

Table 9. Comparison results of F7 test functions in 4 dimensions

Algorithm	Dimension	Minimum Value	Maximum value	Mean	Standard deviation
ACO	2	1.7772	55.2792	25.3269	13.1594
	5	33.0470	133.3064	81.6386	18.5751
	10	88.0778	239.8591	173.6402	35.0953
	30	484.4055	645.3189	544.7705	37.1596
PSO	2	7.322583e-09	1.0012715	0.1596454	0.350144
	5	2.005007	22.881284	8.2732804	4.629569
	10	14.86362	62.029896	34.954967	11.06757
	30	120.8475	264.78830	201.04720	28.58588
WOA	2	1.603E-10	1.992E+00	3.064E-01	5.525E-01
	5	1.062E-05	1.009E+01	4.128E+00	2.697E+00
	10	7.033E-01	4.856E+01	2.281E+01	1.020E+01
	30	1.578E-01	2.215E+02	1.238E+02	6.525E+01
CWOA	2	3.5527E-15	8.0920E-05	3.8136E-06	1.3626E-05
	5	0	1.8157E+00	3.8272E-02	2.5667E-01
	10	1.0658E-13	7.2853E-01	3.2733E-02	1.4381E-01
	30	8.3437E-11	9.2395E+00	3.9202E-01	1.5630E+00
LWOA	2	1.3500E-13	1.9899E+00	2.0225E-01	4.4047E-01
	5	1.3683E-03	9.8713E+00	4.6314E+00	2.5251E+00
	10	3.9005E-02	4.2370E+01	1.9305E+01	1.0999E+01
	30	1.7772	55.2792	25.3269	13.1594
IWOA	2	33.0470	133.3064	81.6386	18.5751
	5	88.0778	239.8591	173.6402	35.0953
	10	484.4055	645.3189	544.7705	37.1596
	30	7.322583e-09	1.0012715	0.1596454	0.350144

Figure 1 to Figure 7 show the results of the adaptation values of the six algorithms under the seven benchmark test functions. It can be found from the curves in the figures. Under different benchmark test functions, with the increasing number of iterations, the algorithms in this paper all show a decreasing trend in the benchmark test functions, while the other five algorithms are trapped in local optimum, which affects the generation of optimal solutions of the algorithms. In particular, in the F7 test function, the algorithm in this paper searches for the theoretical optimum. In the other test functions, although the optimal solution is not searched, the convergence accuracy of the algorithm in this paper is still optimal. In the F1 test function and F2 test function, this paper is able to improve 80 and 40 orders of magnitude respectively, and in the other F3, F4, F5, and F6 test functions, compared with the CWOA algorithm by roughly 2, 14, 8, and 6 orders of magnitude respectively, compared with the LWOA algorithm by roughly 5, 20, 7, and 4 orders of magnitude respectively, compared with the WOA algorithm by roughly 4, 17, and 6, and 5 orders of magnitude respectively. These data results show that the effect of the algorithm improvement in this paper is obvious. From the overall data of the algorithm, the corresponding curve of the algorithm in this paper stabilizes in the second half of some test functions, which indicates that the algorithm falls into the local optimum at the later stage, resulting in the degradation of the performance of the algorithm, which also indicates that there is still much room for improvement of the algorithm in this paper.

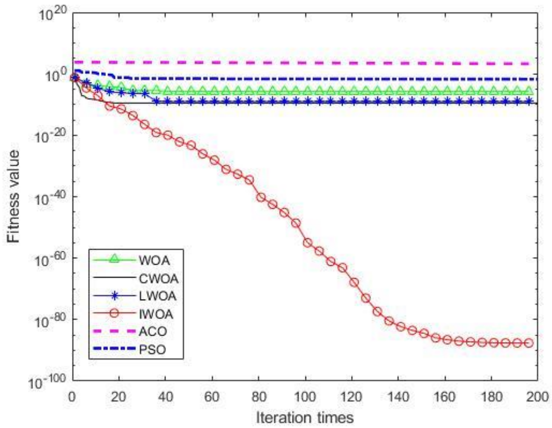

Figure 1. Results of 6 algorithmic fitness values in the F1 function

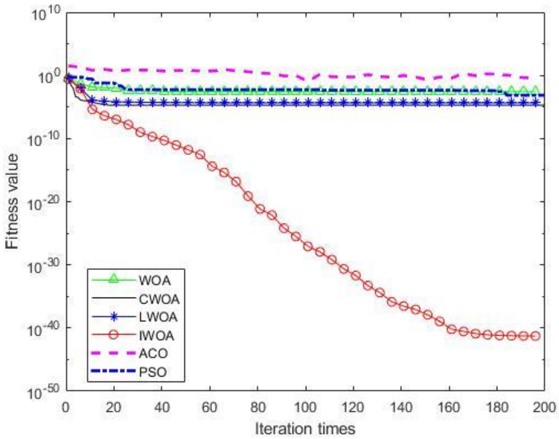

Figure 2. Results of 6 algorithmic fitness values in the F2 function

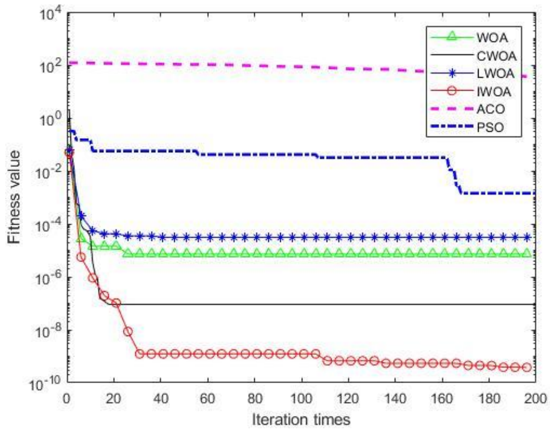

Figure 3. Results of 6 algorithmic fitness values in the F3 function

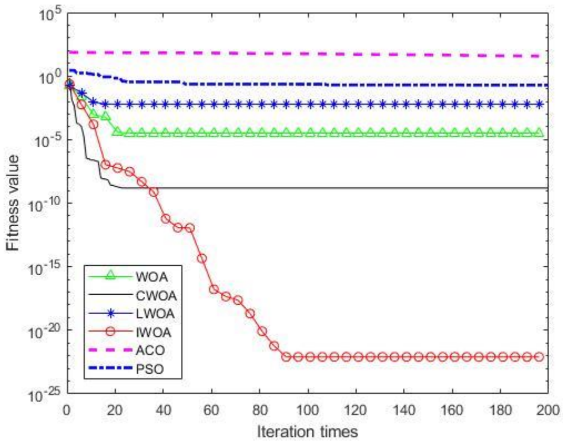

Figure 4. Results of 6 algorithmic fitness values in the F4 function

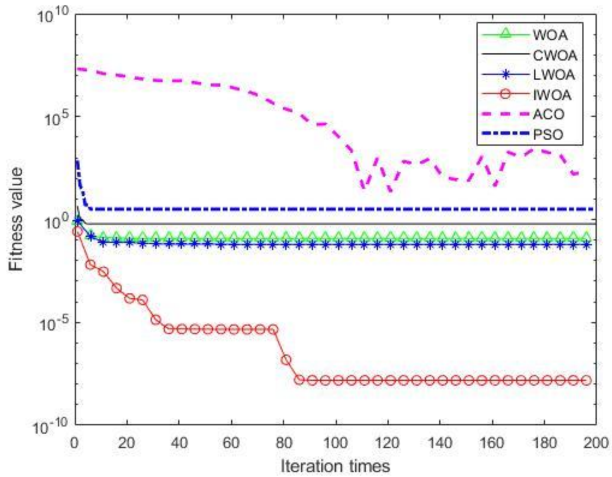

Figure 5. Results of 6 algorithmic fitness values in the F5 function

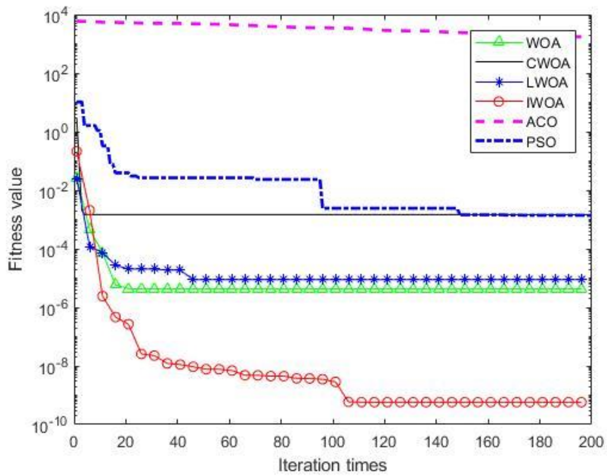

Figure 6. Results of 6 algorithmic fitness values in the F6 function

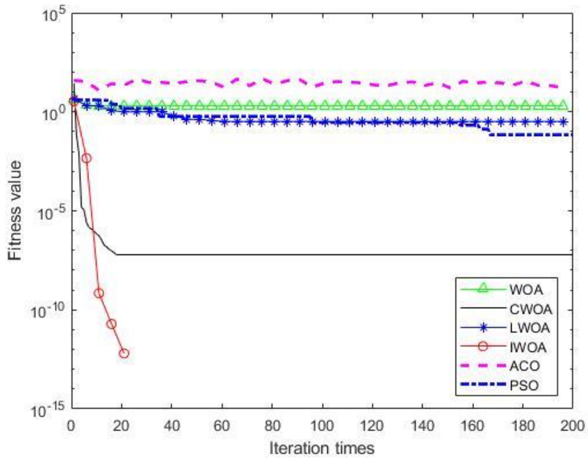

Figure 7. Results of 6 algorithmic fitness values in the F7 function

5.4 Wilcoxon's Test

According to the result of Table 3 to Table 9, the performance of the 6 algorithms can be sorted into the following order: IWOA, CWOA, LWOA, WOA, PSO, ACO. Additionally, due to the importance of the multiple-problem statistical analysis, Table 10 also gives the statistical analysis resultsthroug Wilcoxon's test between IWOA and other 5 compared algorithms. The parameters of Wilcoxon's test are α = 0.01 and 0.05.

Table 10. Wilcoxon's test between IWOA and other algorithms over all dimensions on 7 test functions

Algorithm	R+	R-	P value
IWOA versus ACO	349	236	0.0069
IWOA versus PSO	339	236	0.0091
IWOA versus WOA	289	191	0.0124
IWOA versus CWOA	194	125	0.1282
IWOA versus LWOA	228	143	0.0601

From the results shown in Table 10, we can see that IWOA provides higher R+ values than R- in all the cases. Therefore, we can obtain the conclusions: IWOA is better than CWOA, LWOA, WOA, PSO, ACO significantly.

6. Conclusion

This paper starts from analyzing the basic principle of WOA, and proposes an optimization strategy for the shortcomings of the algorithm in terms of convergence speed and solution accuracy. Chaotic steganography is used in the initialization to improve the diversity of the population; the cosine-based inertia weight method is used in the whale spiral update to ensure that the algorithm does not fall into local optimum, and the Levy behavior is used in the whale foraging behavior to improve the individual global search ability. In the simulation experiments, we choose five different algorithms as the comparison algorithms in this paper, and test them in four technical indicators with different numerical dimensions of seven benchmark functions, and the experimental results show that the performance of the algorithms in this paper has been improved.

Acknowledgements

This work is supported by Henan Provincial Key Scientific Research Project Plan of Colleges and Universities (18B520020).

References

[1] S. Mirjalili S. A. Lewis, The whale optimization algorithm, *Advances in Engineering Software*, Vol. 95, pp. 51-67, May, 2016.

[2] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[3] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters*, Vol. 85, No. 6, pp. 317-325, March, 2003.

[4] M. D. Li, G. H. Xu, Q. Lai, J. Chen, A chaotic strategy-based quadratic Opposition-Based Learning adaptive variable-speed whale optimization algorithm, *Mathematics and Computers in Simulation*, Vol. 193, pp. 71-99, March, 2022.

[5] W. Yang, K. Xia, S. Fan, L. Wang, T. Li, J. Zhang, Y. Feng, A Multi-Strategy Whale Optimization Algorithm and Its Application, *Engineering Applications of Artificial Intelligence*, Vol. 108, Article No. 104558. February, 2022.

[6] S. Chakraborty, A. K. Saha, R. Chakraborty, M. Saha, S. Nama, HSWOA: An ensemble of hunger games search and whale optimization algorithm for global optimization, *International Journal of Intelligent Systems*, Vol. 37, No. 1, pp. 52-104, January, 2022.

[7] S. K. Dewi, D. M. Utama, A new hybrid whale optimization algorithm for green vehicle routing problem, *Systems Science & Control Engineering*, Vol. 9, No. 1, pp. 61-72, January, 2021.

[8] H. Nguyen, X. N. Bui, Y. Choi, C. W. Lee, D. J. Armaghani, A novel combination of whale optimization algorithm and support vector machine with different kernel functions for prediction of blasting-induced fly-rock in quarry mines, *Natural Resources Research*, Vol. 30, No. 1, pp. 191-207, February, 2021.

[9] S. Sharmila, S. Vijayarani, Association rule mining using fuzzy logic and whale optimization algorithm, *Soft Computing*, Vol. 25, No. 2, pp. 1431-1446, January, 2021.

[10] J. Luo, H. Chen, A. A. Heidari, Y. T. Xu, Q. Zhang, C. Y. Li, Multi-strategy boosted mutative whale-inspired optimization approaches, *Applied Mathematical Modelling*, Vol. 73, pp. 109-123, September, 2019.

[11] A. A. Heidari, I. Aljarah, H. Faris, H. L. Chen, J. Luo, S. Mirjalili, An enhanced associative learning-based exploratory whale optimizer for global optimization, *Neural Computing and Applications*, Vol. 32, No. 9, pp. 5185-5211, May, 2020.

[12] G. Kaur, S. Arora, Chaotic Whale Optimization Algorithm, *Journal of Computational Design & Engineering*, Vol. 5, No. 3, pp. 275-284, July, 2018.

[13] M. Abdel-Basset, G. Manogaran, D. El-Shahat, S. Mirjalili, Retracted: A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem, *Future Generation Computer Systems*, Vol. 85, pp. 129-145, August, 2018.

[14] G. L. Sayed, A. Darwish, A. E. Hassanien, A new chaotic whale optimization algorithm for features selection, *Journal of Classification*, Vol. 35, No. 2, pp. 300-344, July, 2018.

[15] N. Zeng, D. Song, H. Li, Y. C. You, Y. R. Liu, F. E. Alsaadic, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, *Neurocomputing*, Vol. 432, pp. 170-182, April, 2021.

[16] J. Zhou, S. Zhu, Y. Qiu, D. J. Armaghani, A. N. Zhou, W. X. Yong, Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm, *Acta Geotechnica*, pp. 1-24, January, 2022.

[17] H. H. Ran, S. P. Wen, K. B. Shi, T. W. Huang, Stable and compact design of Memristive GoogLeNet Neural Network, *Neurocomputing*, Vol. 441, pp. 52-63, June, 2021.

[18] S. P. Wen, J. D. Chen, Y. C. Wu, Z. Yan, Y. T. Cao, Y. Yang, T. W. Huang, CKFO: Convolution kernel first operated algorithm with applications in memristor-based convolutional neural network, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 40, No. 8, pp. 1640-1647, August, 2021.

[19] Y. C. Ao, Y. B. Shi, W. Zhang, Y. J. Li, Improved Particle Swarm Optimization with Adaptive Inertia Weight, *Journal of University of Electronic Science and Technology of China*, Vol. 43, No. 6, pp. 874-880, November, 2014.

[20] A. M. Edwards, R. A. Phillips, N. W. Watkins, M. P. Freeman, E. J. Murphy, V. Afanasyev, S. V. Buldyrev, M. G. E. D. Luz, E. P. Raposo, H. E. Stanley, G. M. Viswanathan, Revisiting Levy flight search patterns of wandering albatrosses, bumblebees and deer, *Nature*, Vol. 449, pp. 1044-1048, October, 2007.

Biographies

Xiaoming Shi received the B.S. degree in computer science and technology from Zhengzhou University of Light Industry, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. His research interests include algorithm design and multi-agent system.

Kun Li received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. His research interests include cloud computing and algorithm design.

Liwei Jia received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. His research interests include cloud computing and algorithm design.