Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **2**, May **2022**

Whale Optimization Algorithm based on Nonlinear Adjustment and Random Walk Strategy

Junfu Xi

Information Engineering Department
Hebei Vocational University of Technology and Engineering
Xingtai 054035, China
Email: beij-08@163.com

Yehua Chen*

Accounting Department
Hebei Vocational University of Technology and Engineering
Xingtai 054035, China
Email: chensmile0318@163.com

Xia Liu and Xiaoji Chen

Information Engineering Department
Hebei Vocational University of Technology and Engineering
Xingtai 054035, China
Emails: 283281266@qq.com and 635909530@qq.com

*Corresponding Author: chensmile0318@163.com

Received September 2021; revised November 2021

Abstract. *In view of the low convergence accuracy, slow convergence speed and easy to fall into local optimum of standard whale algorithm, a whale optimization algorithm based on nonlinear adjustment and random walk strategy (NWOA) is proposed. Based on the exponential function of the maximum fitness value, average fitness value, minimum fitness value and random factor of the population, a strategy adjusting the inertia weight nonlinearly is designed to improve the convergence speed and optimization accuracy of the algorithm. In addition, a nonlinear adjustment convergence factor strategy is used to balance the global search and local development capabilities of the algorithm. A search strategy based on random walk is designed to help the algorithm jump out of the local optimum and improve the local search ability. Through 12 benchmark functions solving, the experimental results show that NWOA algorithm has significantly improved the convergence speed and optimization accuracy, and solved the problem that the algorithm is easy to fall into local optimization in multimodal functions. Thus compared with other swarm intelligence algorithms, NWOA algorithm shows better optimization performance.*
Keywords: whale optimization algorithm, nonlinear adjustment, random walk strategy, function optimization

1. **Introduction.** Optimization problem has always been a research hotspot in computer science, engineering research and other fields. Inspired by the swarm intelligence in nature, a variety of swarm intelligence optimization algorithms, such as Particle Swarm Optimization (PSO) [1], Ant Colony Optimization (ACO) [2], Gey Wolf Optimization (GWO) [3], Ant Lion Optimization (ALO) [4], and Sine Cosine Algorithm (SCA) [5] were proposed. They are widely used in the solution of complex function optimization

problem [6–9], which fully proves the powerful optimization performance of the swarm intelligence optimization algorithm.

Whale Optimization Algorithm (WOA) is a new swarm intelligence optimization algorithm proposed by Mirjalili et al. in 2016 [10]. Through random search, prey surrounding, and bubble-net foraging of humpback whales, it simulates the group hunting behavior of humpback whales and achieves the optimization. The whale optimization algorithm has simple structure and few adjustable parameters. It is faster and more accurate than traditional particle optimization algorithms and genetic algorithms in terms of multi-function solving. Thus WOA algorithm has been widely used in machine learning [11], path planning [12], feature selection [13], production scheduling [14], and image segmentation [15] due to its good optimization performance.

Although WOA is widely used, it also has the disadvantages of traditional swarm intelligence optimization algorithm, such as easy to fall into local optimum, slow convergence speed and low optimization accuracy. Aiming to these shortcomings, many improved algorithms have been put forward to improve the performance of traditional WOA. Oliv et al. presented a hybrid pure WOA [16], which used the hybrid pure strategy to guide the position update of whales, and avoided falling into the local optimum to a certain extent. Zhou et al. introduced Levy flight into the whale optimization algorithm to increase the population diversity and improve the ability to jump out of local optimal solution [17]. Sun et al. presented an improved whale optimization algorithm to solve large-scale global optimization problems. Cosine function, quadratic interpolation and Levy flight strategy were also introduced to avoid falling into local optimum effectively [18]. Based on adaptive weights and simulated annealing, Chu et al. put forward a whale optimization algorithm to effectively improve the convergence accuracy and optimization performance of the algorithm [19]. Liu et al. proposed a WOA with a global search strategy. They introduced adaptive weight and optimal neighborhood disturbance to improve the convergence speed and optimization accuracy [20]. The above improved strategy improves the performance of WOA algorithm in some degree, and there is still room for further improvement.

Aiming at the problems of slow convergence, low optimization accuracy, and easy to fall into local optimum of traditional WOA algorithms, a Nonlinear Adjustment of Inertia Weight and Random Walk Strategies-Based Whale Optimization Algorithm (NWOA) is put forward. In the standard WOA algorithm, the inertia weight in the position update formula is a fixed value of 1. When dealing with complex optimization problems, the search process of the algorithm is extremely complicated, and the obtained solutions are random. Inspired by the PSO algorithm, a strategy for nonlinearly adjusting the inertia weight based on the exponential function of the maximum fitness value, average fitness value, minimum fitness value and random factor of the population is designed. Based on the real-time performance behavior of the population during the iterative process, this strategy is introduced into the position update formula to guide the population to optimize it optimally. Meanwhile, the convergence factor plays a key role in the global development and local search ability of the coordination algorithm, and its setting decreases linearly with the number of iterations from 2 to 0. However, when solving complex optimization problems, the linear decreasing strategy of convergence factor is difficult to adapt to the actual search situation. Therefore, a nonlinear adjusting convergence factor strategy is designed to coordinate the global development and local search ability of the algorithm. At the same time, in the iterative process of optimization, inspired by the ALO algorithm, a search strategy based on random walk is designed. A new solution is obtained by perturbing the contemporary optimal solution through the random walk strategy. Greedy selection strategy is used to compare the fitness values of contemporary optimal solution and new solution, and select and retain the solution with better fitness value. This strategy is conducive to jumping out of local optimization. Meanwhile, the performance of NWOA algorithm is verified by 12 benchmark functions. The results show that the algorithm performs well in function optimization, has higher optimization accuracy and faster convergence speed, and can effectively avoid the algorithm falling into local optimal solution.

2. **Whale optimization algorithm.**

The characteristics of WOA can be described from three aspects: prey surrounding, bubble-net foraging and random search.

Assuming that the whale population is N and the dimension of the problem space to be solved is d, then the corresponding solution of the i-th whale in the dimensional space is $X_i = (x_i^1, x_i^2, \dots, x_i^d)$ and $i = 1, 2, 3, \dots, N$. The position of each whale in the search space represents a feasible solution to the optimization, and the position of the optimal whale corresponds to the global optimal solution.

2.1. **Prey surrounding.**

In the whale algorithm, the optimal whale position is not known primordially during the search. Therefore, it is assumed that the current solution closest to the objective function generated in the iteration is the optimal whale position. Other individuals of the group swim towards the optimal whale position to update their own positions and surround the prey. The position updating formula is as follows:

$$\begin{cases} X(t+1) = X^*(t) - A \cdot D \\ D = |\mathrm{C} \cdot X^*(t) - X(t)| \end{cases} \tag{1}$$

where, t is the current iteration number; X^* is the global optimal whale position vector; X is the current whale position vector, A and C are the coefficient matrixes. Its formula is:

$$A = 2a \cdot r_1 - a \tag{2}$$

$$C = 2r_2 \tag{3}$$

$$a = 2 - 2t/T \tag{4}$$

where, r_1 and r_2 are the random numbers in $[0, 1]$. a is the convergence factor, and it is linearly reduced from 2 to 0 at iteration. T is the maximum number of iterations.

2.2. **Bubble-net foraging.**

Humpback whales also update their position by spiraling up. The position update is:

$$\begin{cases} X(t+1) = X^*(t) + D' \cdot e^{bl} \cdot \cos(2\pi l) \\ D' = |X^*(t) - X(t)| \end{cases} \tag{5}$$

where, b is a constant that defines the shape of logarithmic helix, and l is a random number in $[-1, 1]$.

Humpback whales swim toward their prey spirally and shrink the enclosure to get close to the prey for food. In order to simulate the bubble-net foraging behavior of humpback whales, when $|A| < 1$, humpback whale individual uses the probability 0.5 as the threshold to determine the position update method and select the behaviors of surrounding prey or spiraling. The position update is as follows:

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ X^*(t) + D' \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5 \end{cases} \tag{6}$$

where, p is a random number in $[0, 1]$.

2.3. **Random search.** In the WOA algorithm, in order to improve the global search ability of whales, when $|A| \geq 1$, the whales randomly swim outside the shrinking encirclement to search for prey. In the current whale population, a whale individual is randomly selected as the global optimal solution, and other whale individuals of the population approach it to update their positions. The position update is:

$$X(t+1) = X_{rand}(t) - A \cdot |\mathrm{C} \cdot X_{rand}(t) - X(t)| \tag{7}$$

where, X_{rand} is the randomly selected whale position vector.

3. **Improved whale optimization algorithm (NWOA).**

3.1. **Non-linear adjustment strategy.** (1) Non-linear adjustment of inertia weight Inertial weight is a key factor to balance the global search ability and local optimization ability of the algorithm [1]. Research shows that a larger inertia weight is beneficial for global search and can increase the population diversity, and a smaller inertia weight can improve the local mining of the algorithm and speed up the convergence speed. It can be seen from formulas (6) and (7) that the inertia weight of the position update in the WOA algorithm is a constant 1. Whereas an appropriate inertia weight adjustment strategy can balance the contradiction between global search and local search, thereby improving the algorithm optimization. In the actual optimization process, the iterative evolution of the algorithm is complex and non-linear, and the inertia weight that reduces simply and linearly cannot match the real optimization well. To this end, this paper designs a nonlinear adjustment of inertia weight strategy to adjust the whale position update method. The calculation of the inertia weight in this strategy is based on the real-time performance behavior of the population at iterative. At each iteration, the inertia weight value is adjusted nonlinearly by the exponential function of the maximum fitness value, average fitness value, minimum fitness value and random factor of the population. On the premise of not increasing the empirical selection parameters, the strategy is formalized as shown in formula (8):

$$\omega^t = \frac{f_{avg} - f_{\min}}{f_{\max} - f_{avg}} \cdot e^{-l} \cdot p \quad , \qquad f_{\max} - f_{avg} \neq 0 \tag{8}$$

where, ω^t is the inertia weight of the t-th iteration;

$f_{\max}$, f_{arg} and $f_{\min}$ are the maximum fitness value, average fitness value, and minimum fitness value of the whale population respectively under current iteration number t; l is the random number in $[0, 1]$, and p obeys a uniform distribution on $[0, 1]$.

The position update based on nonlinear adjustment of inertia weight strategy is:

$$\begin{cases} X(t+1) = \omega^t \cdot X^*(t) - A \cdot D \\ D = |\mathrm{C} \cdot X^*(t) - X(t)| \end{cases} \tag{9}$$

$$\begin{cases} X(t+1) = \omega^t \cdot X^*(t) + D' \cdot e^{bl} \cdot \cos(2\pi l) \\ D' = |X^*(t) - X(t)| \end{cases} \tag{10}$$

$$X(t+1) = \begin{cases} \omega^t \cdot X^*(t) - A \cdot D, & p < 0.5 \\ \omega^t \cdot X^*(t) + D' \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5 \end{cases} \tag{11}$$

$$X(t+1) = \omega^t \cdot X_{rand}(t) - A \cdot |\mathrm{C} \cdot X_{rand}(t) - X(t)| \tag{12}$$

It can be seen from the position update formulas shown in formulas (9)-(12) that, compared with the original position update formulas (1), (5), (6) and (7), the position update based on the strategy of non-linear adjustment of inertia weight first comprehensively considers the fitness of the current whale population to improve the algorithm optimization. And then the inertia weight is affected by the exponential function. Compared with

WOA algorithm, the inertia weight decreases faster at the late iteration and improves the local search ability of the algorithm greatly. Meanwhile, affected by the random factor l, the randomness of the value helps to maintain the population diversity. It cannot only make the algorithm have the ability to jump out of the local optimum, but also improves the global search of it.

(2) Nonlinear adjustment of convergence factor
From the WOA algorithm, the coefficient matrix is used to adjust the balance of the global and local search ability of the algorithm, and the value of A mainly depends on the convergence factor a. A large convergence factor can enhance the global search ability and improve the population diversity, so as to avoid falling into the local optimal solution at the initial stage of iteration, and forming premature convergence. A small convergence factor enables the algorithm to have strong local search ability, speeds up the convergence speed and improves the algorithm efficiency. For this reason, in order to solve the slow convergence speed caused by the slow deceleration at the later iteration of WOA, this paper introduces the strategy of nonlinear adjustment of convergence factor without changing the value boundary of a, so that the balance of global and local search ability is ensured and the convergence speed is improved. The formal description of the strategy is:

$$a^t = a^t - a^t \cdot \sin\left(\frac{\pi}{2} \cdot \frac{t}{T}\right) \tag{13}$$

where, a^t is the value of the convergence factor of the t-th iteration, which is nonlinearly reduced from 2 to 0 during the iteration.

After introducing the strategy of non-linear adjusting convergence factor into the NWOA algorithm, the equation for solving the coefficient matrix is changed from formula (2) to formula (13).

3.2. **Random walk strategy.** The Random walk strategy (RWS) used in this paper draws on the idea of the ant random walk model in the ant division optimization algorithm in [13]. The specific steps are as follows:

(1) At each iteration of the algorithm, the current optimal whale position vector X^t is disturbed by a random walk strategy to obtain a whale position vector X^{tr}, the formula is as follows:

$$X_i^{tr} = \frac{(X_i^t - a_i)\cdot(d_i^t - c_i^t)}{b_i - a_i} + c_i^t, i = 1, 2, 3, \dots, d \tag{14}$$

where, a_i and b_i are the minimum and maximum values of the random walk of the i-th dimension variable, respectively; c_i^t and d_i^t are the minimum and maximum values of the t-th iteration of the i-th dimension variable, respectively.

(2) Calculate the fitness $f(X^{tr})$ of the whale position vector X^{tr};

(3) The current optimal whale position vector value X^t and fitness value $f(X^t)$ are updated according to formula (15):

$$X^t = \begin{cases} X^{tr}, & f(X^{tr}) < f(X^t) \\ X^t, & f(X^{tr}) \geq f(X^t) \end{cases}, \quad f\left(X^t\right) = \begin{cases} f(X^{tr}), & f(X^{tr}) < f(X^t) \\ f(X^t), & f(X^{tr}) \geq f(X^t) \end{cases} \tag{15}$$

(4) Output the current optimal whale position vector X^t, and end the RWS algorithm.

3.3. **NWOA algorithm.** The NWOA algorithm is shown as Table 1.

Table 1. Algorithm flow

```
Pseudo code of NWOA Algorithm
Initialize the whales population $X_i (i = 1, 2, ..., n)$
Calculate the fitness of each search agent
$X^*$ =the best search agent
while($t <$ the maximum number of iterations)
  for each search agent
    Update $a, A, C, l$, and $p$
      if1($p < 0.5$)
        if2($|A| < 1$)
          Update the position of the current search agent by Eq.(9)
        else if2($|A| \geq 1$)
          Select a random search agent ($X_{rand}$)
          Update the position of the current search agent by Eq.(10)
        end if2
      else if 1($p \geq 0.5$)
        Update the position of the current search by Eq.(12)
      end if1
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent, Record $X^*$
  for $i = 1$ to $d$ do
    Improve $X^*$ use by Eq.(14)
    Get $X^{**}$
  end for
  Evaluate $X^*$ and $X^{**}$ use by Eq.(15)
  Update $X^*$if there is a better solution
  $t = t + 1$
end while
return $X^*$
```

3.4. **Algorithm complexity analysis.** The whale population is set as N, the dimension of the problem space to be solved as d, the maximum iteration as T, and the time complexity of standard WOA as $O(N \times d \times T)$. The NWOA algorithm introduces non-linear adjustment inertia weights, non-linear convergence factors and random walk strategies on the basis of standard algorithms. Time complexity analysis of NWOA algorithm in this paper:

The strategy of nonlinear inertia weight and convergence factor does not increase the number of cycles and the additional time complexity, and the time complexity is still $O(N \times d \times T)$. The time complexity of the random walk strategy is $O(d \times T)$. The calculation scale of other processes in the algorithm is small and can be ignored. Therefore, the time complexity of the proposed NWOA algorithm is $O(N \times d \times T)$, which is consistent with the time complexity of standard WOA algorithm.

4. **Experiment and analysis.** In order to verify the optimization of NWOA algorithm, this paper selects 12 standard test functions for simulation, and compares them with the Particle Swarm Optimization algorithm [1] (PSO), the Gray Wolf Algorithm [3] (GWO), and the Vertical and Horizontal Crossover Algorithm [21] (CSO), standard WOA algorithm [10] and LWOA [17]. The simulation experiment platform is based on Windows 10

64-bit operating system, 3.0GHz frequency and 16G memory, and Python 3.9.1 software is used for programming.

4.1. **Standard test functions and experimental settings.** For the 12 standard test functions selected in this article, their names, variable ranges and target values are listed in Table 2. Among them, $f_1 - f_5$ are unimodal functions; $f_6 - f_{10}$ are multimodal functions; f_{11} and f_{12} are fixed dimension functions (d is 2); lb and ub are the lower and upper bounds of decision variables, respectively, and $f_{\min}$ is the global optimal solution of the function.

This paper uses the average value (Ave) and standard deviation (std) of the optimization accuracy as the evaluation indexes. The dimensions d of the test function is set as 30, 50 and 100, respectively; the population as 30, and the maximum iterations as 500. The parameters of PSO, GWO, CSO, standard WOA and LWOA are set according to the corresponding references. The parameter setting of NWOA algorithm is the same as that of standard WOA algorithm, and the parameters of random walk strategy introduced in NWOA algorithm are set according to reference [4].

Table 2. Test functions

Name	Function	lb	ub	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	-100	100	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	-10	10	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	-100	100	0
Rosenbrock	$f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	-30	30	0
Quartic	$f_5(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1]$	-1.28	1.28	0
Restrigin	$f_6(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	-5.12	5.12	0
Ackley	$f_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	-32	32	0
Griewank	$f_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	-600	600	0
Zakharov	$f_9(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$	-5	10	0
Alpine	$f_{10}(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i\rvert$	-10	10	0
Drop wave	$f_{11}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	-5.12	5.12	-1
Camel back	$f_{12}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-5	5	-1.0316

4.2. **Experiment and analysis of algorithm performance comparison.** In order to avoid the influence of randomness on the optimization results, the average value (*Ave*) and standard deviation (*Std*) of the optimization accuracy of 6 optimization algorithms are calculated after 50 times of independent operation. reflects the accuracy of the algorithm, and *Std* reflects the algorithm robustness and stability. The relevant statistical results are listed in Table 3, among which the black fonts are the best results.

Table 3. Comparison of optimal performance of 6 algorithms

Function	d	PSO		GWO		CSO	
		Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*
f_1	30	9.57E-02	3.75E-02	4.16E-31	6.43E-31	7.84E-04	8.32E-04
	50	0.7093	0.193493	2.86E-22	3.11E-22	0.198613	0.201729
	100	1.17E+01	2.541354	3.06E-14	2.09E-14	1.34E+01	7.959127
f_2	30	1.881226	0.950153	8.84E-19	5.67E-19	1.10E-04	6.65E-05
	50	7.600576	1.906098	1.24E-13	6.67E-14	2.57E-02	2.05E-02
	100	2.65E+01	2.696883	4.49E-09	1.53E-09	1.793796	0.587376
f_3	30	6.70E+02	1.72E+03	6.71E-06	1.42E-05	6.81E-02	1.21E-01
	50	1.25E+03	1.99E+02	0.182195	0.252752	1.16E+01	3.486827
	100	2.42E+04	1.51E+04	3.93E+02	6.35152	1.21E+03	4.39E+02
f_4	30	7.79E+01	3.35E+01	2.67E+01	0.85153	2.87E+01	4.906496
	50	3.87E+02	4.54E+02	4.72E+01	0.39704	6.52E+01	3.10E+01
	100	1.60E+03	1.60E+02	9.81E+01	0.391692	4.42E+02	1.91E+02
f_5	30	0.1329	0.038805	1.54E-03	8.77E-04	1.55E-03	1.03E-03
	50	0.440187	0.11684	2.31E-03	4.94E-04	5.15E-03	1.31E-03
	100	4.027608	1.189289	7.32E-03	8.70E-04	6.59E-03	3.60E-03
f_6	30	3.50E+01	4.661145	3.01E-14	3.76E-14	5.95E-04	8.23E-04
	50	7.23E+01	1.75E+01	6.851316	2.340851	2.36E-02	8.63E-03
	100	2.92E+02	1.28E+01	8.81378	6.498429	1.01E+02	6.746231
f_7	30	2.972257	0.218551	2.56E-08	1.58E+08	5.05E-02	2.04E-02
	50	2.562834	0.171706	2.26E-06	1.14E-07	0.212828	0.156377
	100	2.67226	6.72E-02	2.45E-04	6.46E-05	1.960989	0.159618
f_8	30	2.22E+02	2.40E+01	6.18E-03	1.65E-03	7.60E-03	1.34E-02
	50	4.21E+02	1.34E+01	7.79E-03	1.42E-02	3.25E-02	1.73E-02
	100	9.17E+02	3.77E+01	1.09E-02	1.94E-02	0.374158	0.168336
f_9	30	4.63E+02	1.79E+02	9.43E-09	1.29E-08	9.00E-02	9.92E-02
	50	1.16E+03	4.71E+02	3.97E-02	3.79E-02	1.06E+01	4.04974
	100	6.53E+03	2.54E+03	7.51E+01	1.97E+01	1.34E+02	7.72E+01
f_{10}	30	0.57209	0.319026	4.55E-04	5.56E-04	3.15E-04	4.35E-04
	50	2.624156	1.688407	6.34E-04	7.37E-04	2.08E-03	1.33E-03
	100	13.44381	3.19191	3.65E-03	3.16E-03	0.208159	0.124683
f_{11}	2	0.999999	1.07E-06	0.999999	2.68E-02	**-1**	**0**
f_{12}	2	**-1.0316**	1.14E-08	**-1.0316**	5.69E-10	**-1.0316**	**0**

Function	d	WOA		LWOA		NWOA	
		Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*
f_1	30	9.72E-20	3.76E-19	1.62E-20	3.44E-20	**0**	**0**
	50	1.22E-15	1.91E-15	4.04E-15	5.89E-16	**0**	**0**
	100	3.49E-12	1.25E-11	3.26E-10	4.60E-10	**0**	**0**
f_2	30	7.66E-14	1.47E-14	3.17E-03	3.84E-13	**0**	**0**
	50	8.89E-12	7.44E-12	3.34E-11	5.24E-11	**0**	**0**

	100	2.78E-09	4.02E-09	6.73E-10	3.17E-10	**0**	**0**
f_3	30	0.192077	0.53866	1.894728	0.235194	**0**	**0**
	50	2.97544	1.71382	4.24E+02	2.97E+02	**0**	**0**
	100	4.29E+01	5.71E+01	7.62E+02	7.09E+02	**0**	**0**
f_4	30	2.72E+01	0.842227	2.71E+01	0.099515	**0.543402**	**0.136793**
	50	4.80E+01	0.790134	4.73E+01	0.527936	**1.016162**	**0.224031**
	100	9.81E+01	0.35415	9.72E+01	0.908728	**3.898943**	**0.639117**
f_5	30	4.82E-04	1.76E-04	1.34E-03	9.48E-04	**1.13E-05**	**3.78E-05**
	50	5.94E-04	1.50E-04	3.75E-03	1.33E-03	**1.44E-05**	**1.43E-04**
	100	1.11E-03	4.86E-04	5.32E-03	5.70E-03	**2.87E-05**	**1.03E-05**
f_6	30	0	0	0	0	**0**	**0**
	50	1.49E-14	1.37E-14	3.66E-13	7.21E-13	**0**	**0**
	100	7.36E-12	9.97E-12	4.16E-11	4.22E-11	**0**	**0**
f_7	30	6.32E-06	6.14E-06	1.79E-05	1.32E-05	**0**	**0**
	50	3.21E-05	1.20E-05	6.03E-05	4.67E-05	**0**	**0**
	100	1.69E-04	8.57E-05	3.43E-04	2.37E-04	**0**	**0**
f_8	30	3.33E-16	1.05E-16	1.11E-16	1.87E-16	**0**	**0**
	50	8.33E-03	2.63E-02	4.35E-03	1.37E-02	**0**	**0**
	100	2.56E-03	8.10E-03	4.71E-03	1.49E-02	**0**	**0**
f_9	30	5.01E-02	7.49E-02	1.31E+02	4.48E+01	**0**	**0**
	50	7.23872	1.27E+01	4.27E+02	8.83E+01	**0**	**0**
	100	7.82E+01	4.71E+01	1.08E+03	1.80E+02	**0**	**0**
f_{10}	30	1.98E-13	4.74E-13	3.56E-13	6.82E-13	**0**	**0**
	50	2.28E-09	6.33E-09	8.29E-06	1.62E-05	**0**	**0**
	100	2.65E-05	5.49E-05	7.67E-04	5.26E-04	**0**	**0**
f_{11}	2	**-1**	**0**	**-1**	**0**	**-1**	**0**
f_{12}	2	**-1.0316**	**0**	**-1.0316**	1.43E-09	**-1.0316**	**0**

From Table 3, for unimodal functions $f_1 - f_3$ and $f_7 - f_{10}$, PSO, GWO, CSO, WOA, LWOA and NWOA algorithms can converge to the theoretical optimal value 0 when $d = 30$, $d = 50$ and $d = 100$, respectively. The standard deviation is 0 with strong optimization ability and stability. The optimal values and theoretical optimal values found by the other 5 algorithms have different deviation. For unimodal functions f_4 and f_5, the results show that there are some deviations between the optimal values obtained by 6 algorithms and the theoretical optimal values. However, the optimization accuracy of NWOA algorithm in these two test functions is more optimal than the other five algorithms, and the standard deviation *Std* is relatively low. For multimodal function f_6, both WOA and NWOA can converge to the theoretical optimal value 0 and the standard deviation *Std* is 0 when the dimension is $d = 30$. However, in the dimensions of $d = 50$ and $d = 100$, only the NWOA algorithm converges to the theoretical optimal value 0, and the standard deviation *Std* is 0. NWOA algorithm shows good optimization ability and stability in low and high dimensions, and the optimal values and theoretical optimal values found by the other four algorithms have different degrees of deviation. For the fixed dimension function f_{11}, CSO, WOA, LWOA and NWOA algorithms can converge to the theoretical optimal value -1.0, and the standard deviation is 0. For the fixed dimension function f_{12}, all the 6 algorithms converge to the theoretical optimal value -1.0316, and the standard deviation *Std* of CSO, WOA and NWOA algorithms is 0.

In view of further reflection on the advantages of NWOA, the convergence curves of 6 algorithms for 12 standard test functions are given in Figure 1-12.

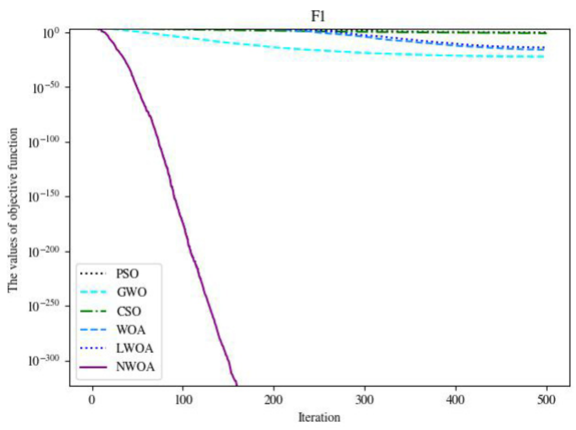

FIGURE 1. $d = 50$, Convergence curve of $f_1(x)$

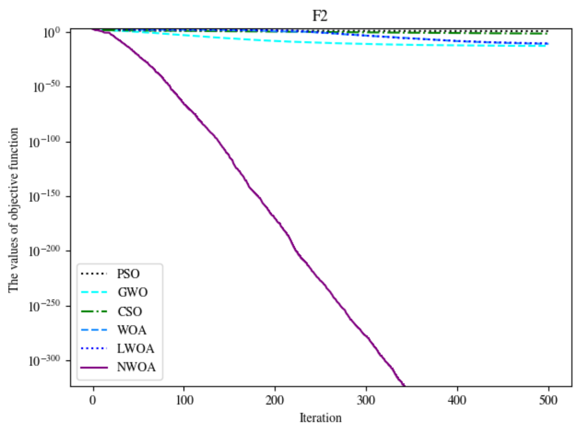

FIGURE 2. $d = 50$, Convergence curve of $f_2(x)$

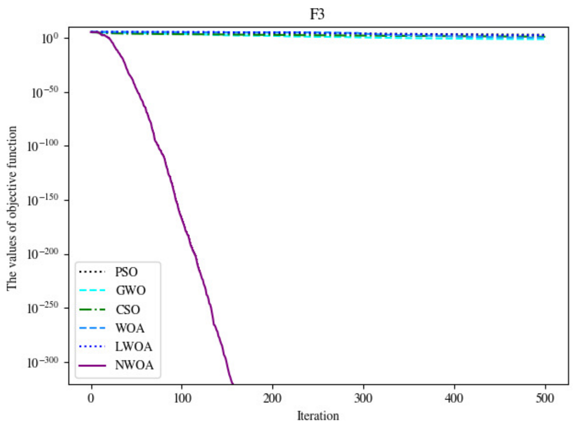

FIGURE 3. $d = 50$, Convergence curve of $f_3(x)$

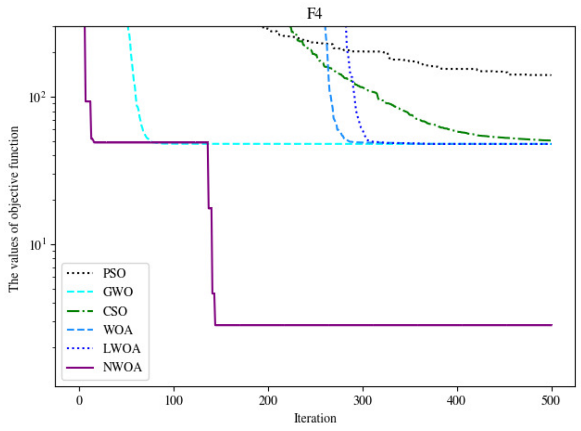

FIGURE 4. $d = 50$, Convergence curve of $f_4(x)$

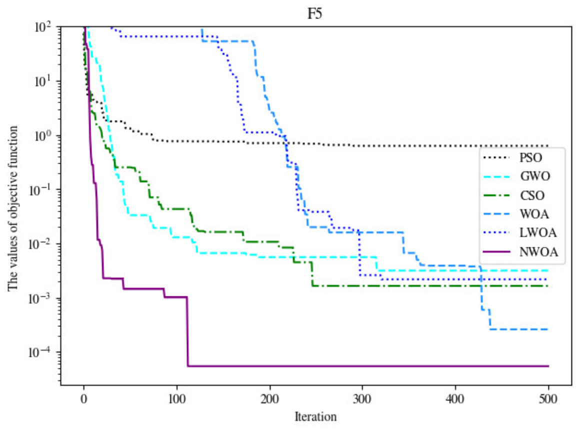

FIGURE 5. $d = 50$, Convergence curve of $f_5(x)$

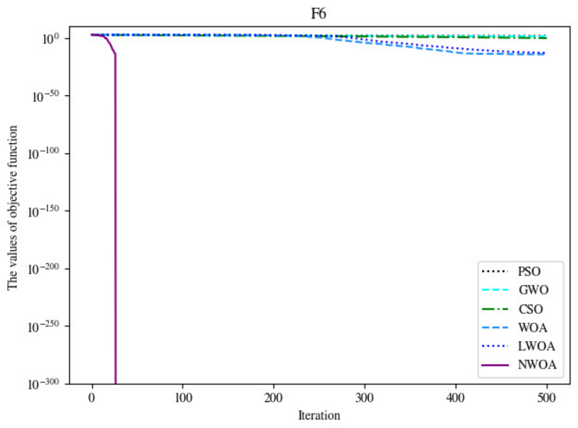

FIGURE 6. $d = 50$, Convergence curve of $f_6(x)$

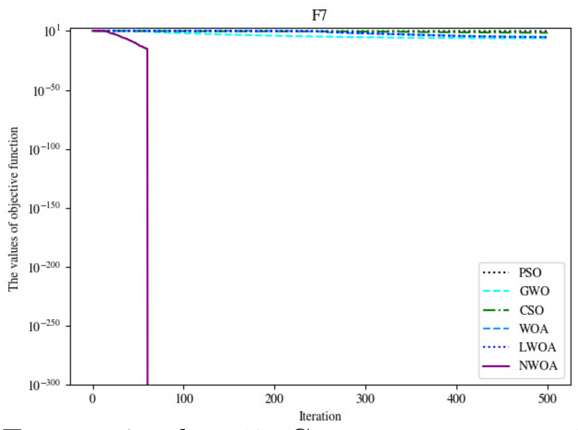

FIGURE 7. $d = 50$, Convergence curve of $f_7(x)$

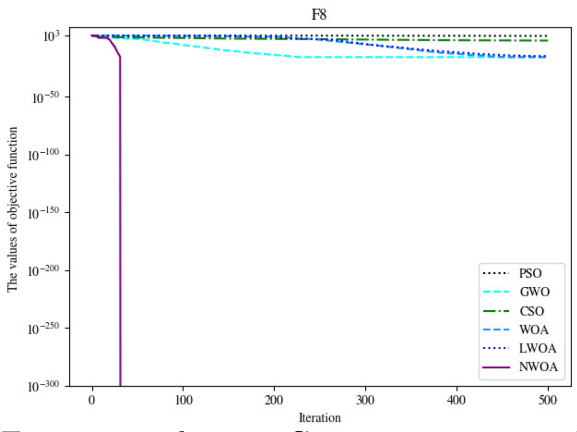

FIGURE 8. $d = 50$, Convergence curve of $f_8(x)$

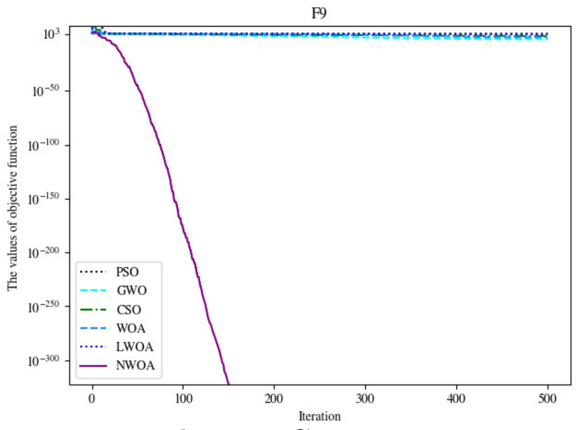

FIGURE 9. $d = 50$, Convergence curve of $f_9(x)$

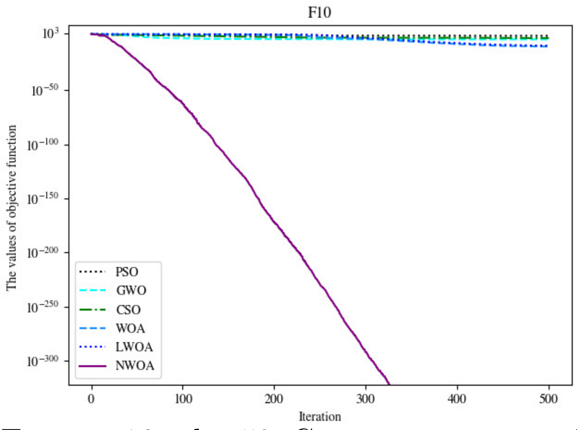

FIGURE 10. $d = 50$, Convergence curve of $f_{10}(x)$

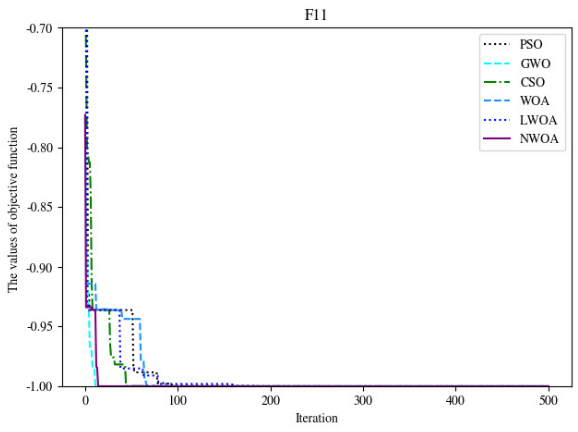

FIGURE 11. $d = 50$, Convergence curve of $f_{11}(x)$

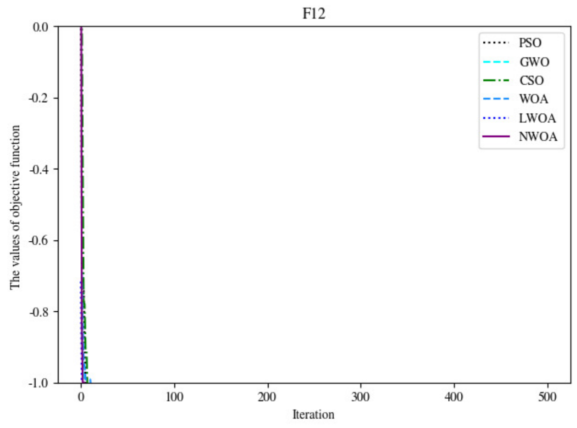

FIGURE 12. $d = 50$, Convergence curve of $f_{12}(x)$

From Figure 1-Figure 10, that the convergence speed of NWOA algorithm is more optimal than the other five algorithms to some extent. For unimodal functions, NWOA has optimal convergence accuracy and speed. For multimodal functions, NWOA algorithm is more resistant to the attraction of regional mechanisms, has the ability to jump out of the local optimal solution, and finally converges to the theoretical optimal solution. From Figure 11-Figure 12, for a fixed 2-dimensional function, the six algorithms can converge or approximate to the theoretical optimal solution. Among them, the GWO and NWOA algorithms perform well in convergence speed.

Generally, the optimization of NWOA algorithm on the single-peak and multi-peak test functions is significantly better than the PSO, GWO, CSO, WOA and LWOA algorithms, and the convergence speed on the fixed 2-dimensional function is slightly better than that of PSO, CSO, WOA and LWOA algorithms. This is mainly due to the nonlinear adjustment of inertia weight, nonlinear adjustment of convergence factors and dynamic walk strategies. These improved strategies can effectively balance the global exploration and local search ability , accelerate the convergence speed, and avoid the algorithm falling into local optimization, so that the algorithm shows good performance in the optimization accuracy, convergence speed and stability.

5. **Conclusion.** Aiming at the slow convergence speed and easy to fall into local optimal solution of WOA algorithm, a whale optimization algorithm based on nonlinear adjustment and random walk strategy (NWOA) is proposed in this paper. First of all, according tothe particle swarm optimization algorithm, the NWOA algorithm introduces the strategy of non-linear adjustment of inertia weight to adjust the whale position update method, and effectively balance the global exploration and local mining capabilities. Considering the fitness of whale population, the optimization accuracy is improved, and the convergence speed and global exploration ability are improved under the random exponential function. Secondly, based on the influence of the convergence factor a in the optimization process, and without changing the value boundary of a, nonlinear adjustment of convergence factor is introduced into the NWOA algorithm to ensure the balance of the global and local search ability of the algorithm, and improve the convergence speed of it. Finally, during the optimization, inspired by random walk strategy in ant lion optimization algorithm, the random walk strategy is introduced into NWOA algorithm to improve the local optimization ability of the algorithm. Through the statistical results of 6 algorithms in 12 test functions, the NWOA algorithm has been significantly improved in terms of convergence speed and optimization accuracy. The problem of easy to fall into the local optimum in the multimodal function is solved with strong optimization. The next research will apply the NWOA algorithm to large-scale function optimization, constrained optimization and practical engineering.

Acknowledgment. The work was supported by Xingtai science and technology planning project (2020ZZ048).

REFERENCES

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[2] M. Dorigo, V. Maniezzo, A. Colomi, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, 1996.

[3] S. Colomi, S.M. Mirjalili, A. Lewis, Grey Wolf optimizer, *Advances in Engineering Software*, vol. 69, no. 3, pp. 46-61, 2014, https://doi.org/10.1016/j.advengsoft.2013.12.007

[4] S. Mirjalili, The ant lion optimizer, *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015, https://doi.org/10.1016/j.advengsoft.2015.01.010

[5] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, *Knowledge-Based Systems*, vol. 96, pp. 120-133, 2016, 96:120-133. https://doi.org/10.1016/j.knosys.2015.12.022

[6] L.L. Kang, R.S. Chen, N.X. Xiong, Y.C. Chen, Y.X. Chen, C.M. Chen, Selecting hyper-parameters of Gaussian process regression based on non-inertial particle swarm optimization in internet of things, *IEEE Access*, vol. 7, pp. 59504-59513, 2019.

[7] Q.Y. Yang, S.C. Chu, J.S. Pan, C. M. Chen, Sine cosine algorithm with multigroup and multistrategy for solving CVRP, *Mathematical Problems in Engineering*, vol. 2020, 2020, https://doi.org/10.1155/2020/8184254

[8] D. Čakmak, Z. Tomičević, H. Wolf, Ž. Božić, H2 optimization and numerical study of inerter-based vibration isolation system helical spring fatigue life, *Archive of Applied Mechanics*, vol. 89, no. 7, pp. 1221-1242, 2019, https://doi.org/10.1007/s00419-018-1495-2

[9] S.D. Li, X.M. You, S. Liu, Multiple ant colony optimization using both novel LSTM network and adaptive Tanimoto communication strategy, *Applied Intelligence*, vol. 51, pp. 5644-5664, 2021, https://doi.org/10.1007/s10489-020-02099-z

[10] S. Mirjalili, A. Lewis, The Whale optimization algorithm, *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016, https://doi.org/10.1016/j.advengsoft.2016.01.008

[11] D.D. Kong, Y.J. Chen, N. Li, C.Q. Duan, L.X. Lu, D.X. Chen, Tool wear estimation in end milling of Titanium alloy using NPE and a Novel WOA-SVM model, *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 5219-5232, 2020.

[12] M. Petrović, Z. Miljković, A. Jokić, A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm, *Applied Soft Computing*, vol. 81, pp. 105520, 2019, https://doi.org/10.1016/j.asoc.2019.105520

[13] M. Shuaib, S.I.M. Abdulhamid, O.S. Adebayo, O. Osho, I. Idris, J.K. Alhassan, N. Rana, Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification, *SN Applied Sciences*, vol. 1, no. 5, pp. 1-17, 2019, https://doi.org/10.1007/s42452-019-0394-7

[14] M. Liu, X.F. Yao, Y.X. Li, Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems, *Applied Soft Computing*, vol. 87, pp. 105954, 2020, https://doi.org/10.1016/j.asoc.2019.105954

[15] E.A.M. Abd, A.A. Ewees, A.E. Hassanien, Whale optimization algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, *Expert Systems with Applications*, vol. 83, pp. 242-256, 2017, https://doi.org/10.1016/j.eswa.2017.04.023

[16] D. Oliva, E.A.M. Abd, A.E. Hassanien, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, *Applied Energy*, vol. 200, pp. 141-154, 2017, https://doi.org/10.1016/j.apenergy.2017.05.029

[17] Y. Ling, Y. Zhou, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, *IEEE Access*, vol. 5, pp. 6168-6186, 2017.

[18] Y.J. Sun, X.L. Wang, Y.H. Chen, Z.J. Liu, A modified whale optimization algorithm for large-scale global optimization problems, *Expert Systems with Applications*, vol. 114, pp. 563-577, 2018, https://doi.org/10.1016/j.eswa.2018.08.027

[19] D.L. Chu, H. Chen, X.G. Wang, Whale optimization algorithm based on adaptive weight and simulated annealing, *Acta Electronica Sinica*, vol. 47, no. 5, pp. 992-999, 2019.

[20] L. Liu, K.Q. Bai, Z.H. Dan, S. Zhang, Z.G. Liu, Whale optimization algorithm with global search strategy, *Journal of Chinese Computer Systems*, vol. 41, no. 9, pp. 1820-1825, 2020.

[21] A.B. Meng, Y.C. Chen, H. Yin, S.Z. Chen, Crisscross optimization algorithm and its appl, *Knowledge-Based Systems*, vol. 67, no. 4, pp. 218-229, 2014.