Improved Whale Optimization Algorithm Based on Nonlinear Adaptive Weight and Golden Sine Operator

J. ZHANG¹ AND J. S. WANG², (Member, IEEE)
¹School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114000, China
²School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114000, China
Corresponding author: J. S. Wang (wang_jiesheng@126.com)

This work was supported in part by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province under Grant 2017FWDF10, and in part by the Project by Liaoning Provincial Natural Science Foundation of China under Grant 20180550700.

ABSTRACT Whale optimization algorithm (WOA) is a swarm intelligence-based algorithm that simulates whale population predation in the sea. Aiming at the shortcomings of WOA such as low precision and slow convergence speed, an improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator (NGS-WOA) was proposed. NGS-WOA first introduced a non-linear adaptive weigh so that search agents can adaptively explore the search space, and balance the development and exploration stages. Secondly, the improved golden sine operator is incorporated into the WOA. Due to the special relationship between the sine function and the unit circle, traversing the sine function is equivalent to scanning the unit circle. The search agent performs an efficient search with a sine route so as to improve the convergence speed and global exploration capability of the algorithm. At the same time, the addition of the golden section coefficient allows search agents to exploit with a fixed shrink step. The search agent can develop to areas with excellent results, which improves the optimization accuracy and local exploitation ability of the algorithm. In the simulation experiments, the gold sine algorithm (GoldSA), whale optimization algorithm (WOA), particle swarm optimization (PSO) algorithm, firefly algorithm (FA), fireworks algorithm (FWA), sine cosine algorithm (SCA) and NGS-WOA were selected for comparison experiments. Then, the effectiveness of the proposed improved strategies is verified. Finally, the improved WOA is applied to high-dimensional optimization and engineering optimization problems. The experimental results show that the improved strategy can effectively improve the performance of the algorithm, so that NGS-WOA has the advantages of high global convergence and avoiding falling into local optimal values.

INDEX TERMS Whale optimization algorithm; nonlinear adaptive weight; golden sine operator; optimization.

I. INTRODUCTION

Meta-heuristic algorithms have been used in more and more engineering problems over the past two decades. Meta-heuristic algorithms are inspired by simple principles in nature and do not require gradient information. They are easy to implement and can be used to solve a wide range of problems covering different areas. Generally, meta-heuristic algorithms simulate natural phenomena to build mathematical models. The meta-heuristic algorithms mainly include evolution-based algorithms, swarm intelligence-based algorithms, and physical phenomenon-based algorithms.

The evolution-based algorithms are inspired by Darwin’s theory of evolution and retain the best individuals from the population as the offspring to perform operations. Evolution-based algorithms include Genetic Algorithm (GA) [1], Evolutionary Strategy (ES) [2], Differential Evolutionary Strategy (DE) [3], Biogeographic-Based Optimization Algorithm (BBO) [4], Probability-Based Incremental Learning (PBIL) [5], etc. The swarm intelligence based algorithms simulate the intelligent behavior of biological swarms (foraging methods, migration routes, mating choices, and information sharing mechanisms) to establish mathematical models.
Such algorithms mainly include Particle Swarm Optimization (PSO) [6], Fireflies Algorithm (FA) [7], Bat Algorithm (BA) [8], Artificial Fish Swarm Algorithm (AFSA) [9], Bird Mating Optimization Algorithm (BMO) [10], Gray Wolf Optimization Algorithm (GWO) [11], Naked Mole-Rat algorithm (NMR) [12], Cuckoo Search (CS) [13], Salp Swarm Algorithm (SSA) [14], Grasshopper Optimization Algorithm (GOA) [15], Moth-Flame Optimization Algorithm (MFO) [16], Ant Lion Optimization Algorithm (ALO) [17], Dolphin Echo Localisation Algorithm [18], Fruit fly Optimization algorithm (FOA) [19], Hunting Search (HS) [20], Krill Herd (KH) [21], Wasp Swarm Algorithm (WSA) [22], Bee Collecting Pollen Algorithm (BCPA) [23], Monkey Search (MS) [24], etc. Physical phenomenon-based algorithms are inspired by physical phenomena, including Black Hole Algorithm (BH) [25], Center Force Optimization Algorithm (CFO) [26], Gravitational Search Algorithm (GSA) [27]. In addition, there are some algorithms inspired by life or human social behavior, such as the Fireworks Algorithm (FWA) [28], Imperialist Competition Algorithm (ICA) [29], Soccer League Competition Algorithm (SLC) [30]–[32], Brainstorming Optimization (BSO) [33], Electromagnetic Optimization Algorithm (EMO) [34], Bacterial Foraging Algorithm (BFA) [35], Anarchic Society Optimization (ASO) [36], Social Based Algorithm (SBA) [37], Taboo Search (TS) [38], etc. In general, swarm intelligence-based algorithms are better than evolution-based algorithms. First, evolution-based algorithms do not retain search space information after preserving the next generation. For example, genetic algorithms form a new generation of populations through selection, crossover, and mutation. Information exchange mechanisms are not used between populations. However, swarm intelligence-based algorithms usually retain information in the search space. For example, the dragonfly algorithm uses the concept of neighborhoods so that individual dragonflies can move and forage based on each other’s information. Second, swarm intelligence-based algorithms typically control fewer parameters and operators than evolution-based algorithms. SI-based algorithms have higher flexibility and can be applied to optimization problems in different disciplines and situations. It is worth mentioning that heuristic Mathematical Programming (MP) techniques based algorithms are relatively novel algorithms that have been proposed in recent years. At present, there is little related literature: the Basic Optimization Algorithm (BOA) [39], [40] uses basic mathematical operators and displacement parameters pointing to the optimal value for optimization. The Sine Cosine Algorithm (CSA) [41] builds a mathematical model by adaptively and equally using sine and cosine search methods. The Gold Sine Algorithm (GoldSA) [42] was inspired by the sine function in trigonometric functions. GoldSA uses the sine function route combined with the golden section coefficient to solve the optimization problem.

The whale optimization algorithm (WOA) [43] is a swarm intelligence-based algorithm proposed by Seyedali Mirjalili in 2016 to solve the optimization problem by establishing a mathematical model that simulates the behavior of whale predation. It relies on simple concepts and contains fewer operators, so it has attracted the attention of many scholars. However, WOA still has defects in solving optimization problems. These limitations inspired scholars to conduct in-depth research on it and proposed many improved methods and practical applications. Majid.Mafarja et al designed two hybrid models based on WOA for feature selection problems [44]. In the first model, the simulated annealing (SA) algorithm is embedded in the WOA. In the second model, the simulated annealing (SA) algorithm is used to improve the best solution found by each iteration of the WOA. Experimental results show that both hybrid WOA can improve classification accuracy. However, the effect of the improved algorithm on the benchmark function has not been demonstrated. In the problem of determining the optimal threshold for image segmentation, the traditional method based on Otsu and Kapur is applicable to the two-level threshold problem. However, determining the optimal threshold in the case of multiple levels of thresholds has the disadvantage of being time-consuming. Mohamed Abd EIAziz et al used WOA to determine the optimal threshold for image segmentation [45], which made up for the time-consuming disadvantage. However, the performance of the improved algorithm in terms of convergence speed and optimization accuracy is unknown. DiegoOliva et al used chaotic mapping to calculate and automatically adjust the internal parameters of WOA, and proposed the parameter estimation problem of the WOA applied to photovoltaic cells [46]. Experiments show that the proposed algorithm improves the ability to find the best solution. At the same time, the robustness of the algorithm has also been effectively improved. However, the effectiveness of the improved algorithm requires more experiments to verify. Jianzhou Wang et al proposed a multi-objective whale optimization algorithm (MOWOA) applied to wind speed prediction system for wind power generation [47], which is used to optimize the weights and thresholds of the Elman neural network used in the prediction system, which has better prediction accuracy and stability. However, the effectiveness of the MOWOA requires more experiments to verify. Salgotra, Rohit et al used the concept of opposite learning, exponentially decreasing parameters, and the concept of eliminating or re-initializing the worst particles to improve the whale algorithm [48]. Experimental results show that the performance of all proposed versions is better than the original WOA. At the same time, the influence of dimensions and variable population on the algorithm is analyzed. WOA has strong ability to solve different optimization problems, but still has the disadvantages of low optimization accuracy, slow convergence speed, and easy to fall into local optimal values. At the same time, according to the No Free Lunch Theorem (NFL) [49], no algorithm can
solve all optimization problems. Improved algorithms may outperform other algorithms on some optimization problems. Therefore, the innovation and necessity of this paper have been strongly supported.

The structure of WOA algorithm is simple, and it has strong optimization ability. However, the shortcomings of WOA are also obvious. First, the location update mechanism of WOA is determined by random parameters, which brings randomness to the algorithm. Randomness allows the algorithm to explore more areas. However, random selection of mathematical models may appear that the selected model is not optimal. At the same time, the spiral update mechanism may trap search agents in a logarithmic spiral, which reduces population diversity. Finally, the location update mechanism has designed parameters that balance the exploitation and exploration phases. However, the parameters are random, which brings uncertainty to the algorithm. In view of the shortcomings of the WOA algorithm, this paper proposes an improved WOA based on nonlinear adaptive weight and golden sine operator (NGS-WOA). The nonlinear weight can be adaptively changed with the iterative process, and guide the search agent to explore the search space. It can find suitable transition between local exploitation and global exploration. Due to the special relationship between the sine function and the unit circle, traversing the sine function is equivalent to scanning the unit circle. The improved golden sine operator enables the search agent to efficiently scan the search space with a sinusoidal route, and improves the global exploration capability of the algorithm. At the same time, the addition of the golden section coefficient makes the search agent move closer to a region with good results with a fixed contraction, which improves the local exploitation ability of the algorithm. In addition, the golden sine operator is used to update the optimal solution found in each iteration of WOA, which eliminates the randomness of the algorithm. The sections of this paper are arranged as follows. Section II introduces the whale optimization algorithm. Section III introduces the improved strategy and the detailed steps of NGS-WOA. Section IV discussed the simulation experiments and analysis of NGS-WOA. First, the improved WOA is compared with other algorithms. The effectiveness of the improved strategy was subsequently verified. Finally, the improved WOA is applied to high-dimensional optimization and engineering optimization problems. Section V is the conclusion of this paper.

II. WHALE OPTIMIZATION ALGORITHM

A. MATHEMATICAL MODEL OF WHALE OPTIMIZATION ALGORITHM

Whales are intelligent animals [50], and their predation process is peculiar. In order to transform the process of whale predation into a mathematical model, WOA regards the optimal value in the search space as a prey. The search agent continuously explores the search space through the location update mechanism, and finally moves closer to the area where the optimal value is located. In order to further simulate the process of whale predation, WOA designs two mathematical models, namely the shrinking enclosing mechanism and the spiral update mechanism. The use probability of these two models is equal. To implement the mathematical model, the algorithm first generates a random number \( p \) in \([0,1]\). If \( p < 0.5 \), choose the shrinking enclosing mechanism. If \( p \geq 0.5 \), choose the spiral update mechanism. The shrinking enclosing mechanism and the spiral update mechanism are described as follows.

\[
\tilde{X}(t + 1) = \begin{cases} 
\tilde{X}_p(t) - \hat{A} \cdot \left| \hat{C} \cdot \tilde{X}_p(t) - \tilde{X}(t) \right| & \text{if } p < 0.5 \\
\tilde{D} \cdot e^{bl} \cdot \cos(2\pi t) + \tilde{X}_p(t) & \text{if } p \geq 0.5 
\end{cases}
\] (1)

The shrinking enclosing mechanism can simulate the process of whale predation, which is described as follows:

\[
\tilde{X}(t + 1) = \tilde{X}_p(t) - \hat{A} \cdot \left| \hat{C} \cdot \tilde{X}_p(t) - \tilde{X}(t) \right| 
\] (2)

where, \( \tilde{X} \) indicates the position vector of the search agent, \( t \) is the current number of iterations, \( \tilde{X}_p(t) \) indicates the position of the current optimal value (prey), \( \hat{A} \cdot \left| \hat{C} \cdot \tilde{X}_p(t) - \tilde{X}(t) \right| \) indicates the indentation distance of the search agent to the current optimal value, which is used to simulate the process of the whale approaching the prey, \( \hat{A} \) and \( \hat{C} \) are coefficient vectors used to control the movement effect of the search agent, which can be described as:

\[
\hat{A} = 2\bar{a} \cdot \tilde{r} - \bar{a} \\
\hat{C} = 2 \cdot \tilde{r} 
\] (3)

(4)

where, \( \bar{a} \) is a random number in \([-1,1] \), \( \bar{a} \) indicates the convergence factor, which gradually decreases from 2 to 0 as iterations, \( \tilde{r} \) is a random number in \([0,1] \). It is not difficult to see that the shrinking enclosing mechanism with the decrease of the convergence factor \( \bar{a} \) can realize the continuous shrinking process of whales.

In fact, in addition to shrinking and enclosing the prey, whale populations will also randomly look for prey based on the location of the individual. The equation for random search of whale population is described as:

\[
\tilde{X}(t + 1) = \tilde{X}_{\text{rand}}(t) - \hat{A} \cdot \left| \hat{C} \cdot \tilde{X}_{\text{rand}}(t) - \tilde{X}(t) \right| 
\] (5)

where, \( \hat{A} \) has the same meaning as Eq (2), \( \tilde{X}_{\text{rand}}(t) \) indicates a randomly selected position vector from the search agent. It can be seen from Eq (5) that the search agent can realize the process of whale randomly finding prey by selecting random vectors.

The shrinking enclosing mechanism can simulate the behavior of whale populations enclosing prey and randomly searching for prey. To implement this mathematical model, the shrinking enclosing mechanism switches the search mode through the coefficient vector \( \hat{A} \). The model of the shrinking enclosing mechanism is shown in Fig. 1. Among them, \((x^*, y^*)\) indicates the position of the current search agent, and \((x^*, y^*)\)
indicates the position of the current optimal value. Fig. shows that the coefficient vector $\vec{A}$ can guide search agents through the exploitation or exploration process. When $|\vec{A}| \leq 1$, the search agent can move closer to the current optimal value, which reflects the local exploitation ability of the algorithm. When $|\vec{A}| > 1$, the search agent can randomly search the area outside the current optimal value, which reflects the global exploration ability of the algorithm. The shrinking enclosing mechanism closes or spreads the optimal value area, which makes the algorithm well balance the exploitation and exploration.

It can be known from Eq (1) that, in addition to the shrinking enclosing mechanism, another mathematical model of the whale optimization algorithm is the spiral update mechanism, which can be described as:

$$\vec{X}_t \rightarrow \vec{X}_{t+1} = \vec{D} \cdot e^{bl} \cdot \cos (2\pi l) + \vec{X}_P (t) \quad (6)$$

where, $b$ is a constant used to limit the shape of the logarithmic spiral, $l$ is a random number in $[-1,1]$, which is used to control the indentation effect of the search agent. $\vec{D}$ indicates the distance between the $i$-th search agent whale and the current optimal value, which can be calculated by:

$$\vec{D} = \vec{X}_P (t) - \vec{X} (t) \quad (7)$$

The model of the spiral update mechanism is shown in Fig. 2.

It can be seen from Fig. 2 that the spiral update mechanism first calculates the distance between the search agent and the current optimal value. Then, a logarithmic spiral route is created between them to ensure that the search agent moves closer to the target value. In addition, Fig. 2 shows the effect of parameter $l$ on the indentation effect of search agents. The decrease of parameter $l$ can enhance the indentation effect of the search agent to the current optimal value, and increase the local search ability of the algorithm. Increasing the
parameter $l$ can weaken the indentation effect of the search agent to the current optimal value, and reduce the probability of the algorithm falling into a local optimum.

**B. PSEUDO CODE AND DISCUSSION OF WOA ALGORITHM**

The pseudo code of WOA is described as follows. Fig. 3 is the flowchart of WOA.

- Initialize the whales population $X_i$ ($i = 1, 2, 3 \ldots n$)
- Calculate the fitness of each search agent $X_p$ = the best search agent
- $t = 1$
- While ($t < \text{maximum number of iterations}$)
  - For each search agent
    - Update $a, A, C, l$, and $p$
      - If ($p < 0.5$)
        - If ($|A| < 1$)
          - Update the position of search agent by the Eq.(2)
        - Else If ($|A| \geq 1$)
          - Update the position of search agent by the Eq.(5)
      - Else If ($p > 0.5$)
        - Update the position of the search agent by the Eq.(6)
  - Check if any search agent goes beyond the search space and amend it
  - Update $X_p$ if there is a better solution
  - $t = t + 1$
- End While
- Return $X_p$

From the pseudo code and mathematical model, it can be seen that WOA has the following advantages: 1) The concept and structure of the algorithm are simple and easy to implement. 2) WOA regards the current best solution as prey. The search agent updates the location according to the prey, and can conduct exploration and exploitation without losing the target value, and develop to an excellent area in the search space. 3) WOA retains the best solution for each iteration, even if the quality of the entire population declines, it does not affect the prey. 4) The position update mechanism designed by WOA considers the balance between development and exploration, and can avoid falling into local optimal values. 5) As a swarm intelligence algorithm, WOA will continuously improve the initialized random solution. Compared with individual-based algorithms, it has better global exploration.

At the same time, according to the discussion about WOA in the introduction, we should also consider the potential flaws of WOA: 1) Too many random parameters bring uncertainty to the algorithm. 2) WOA designed the parameters to balance the exploitation and exploration phases. However, the parameters are random. The parameters should change adaptively with iteration. 3) WOA algorithm is easy to fall into local optimal value on multi-peak function. Improvement strategies need to be introduced to further enhance the transition between exploitation and exploration.

**III. IMPROVED WHALE OPTIMIZATION ALGORITHM BASED ON NONLINEAR ADAPTIVE WEIGHT AND GOLDEN SINE OPERATOR**

**A. NONLINEAR ADAPTIVE WEIGHT**

Local exploitation and global exploration are two important stages of the optimization algorithm. Excessive exploitation capacity will inhibit the tendency of search agents to develop towards the global optimal solution. Excessive exploration capabilities will reduce the optimization accuracy. Therefore, finding an appropriate transition between the two stages can improve the performance of the optimization algorithm. WOA has designed two mathematical models, namely the shrinking enclosing mechanism and the spiral update mechanism. In the shrinking enclosing mechanism, when $|A| \leq 1$, the search agent moves closer to the area where the target value is located, which reflects the exploitation ability. When $|A| > 1$, the search agent randomly searches the area outside the target value, which reflects the exploration ability. In the
spiral update mechanism, the decrease of parameter \( l \) can enhance the indentation effect of the search agent to the target value, which reflects the exploitation ability. The increase of parameter \( l \) can weaken the indentation effect of the search agent to the target value. This process reduces the probability of the algorithm falling into a local optimum and reflects the exploration capability. It can be seen that the changes of parameters \( \bar{A} \) and \( l \) directly affect the balance between exploitation and exploration, and then affect the performance of the algorithm. However, the changes of parameters \( \bar{A} \) and \( l \) are random, which brings randomness to the algorithm. Therefore, this paper introduces a nonlinear adaptive weight, so that parameters \( \bar{A} \) and \( l \) can be adaptively adjusted with the number of iterations so as to reduce the impact of randomness on the performance of the algorithm. The proposed nonlinear adaptive weight \( C_1 \) are described by the following equation:

\[
C_1 = \begin{cases} 
\frac{1}{2} \cdot \left[ 1 + \cos \left( \frac{\pi t}{T} \right) \right]^k & \text{if } t < \frac{T}{2} \\
\frac{1}{2} \cdot \left[ 1 - \cos \left( \pi + \frac{\pi t}{T} \right) \right]^k & \text{if } t \geq \frac{T}{2}
\end{cases}
\]  

where, \( t \) is the current number of iterations, \( T \) is the total number of iterations, \( k \) indicates the adjustment factor, which is used to adjust the convergence effect of the nonlinear adaptive weight. The influence of the adjustment factor \( k \) on the nonlinear adaptive weight \( C_1 \) is shown in Fig. 4. It can be seen from Fig. 4 that when \( k = 1/2 \), the nonlinear adaptive weight have the best convergence effect. The nonlinear adaptive weight has a slow downward trend in the early stage and a significant downward trend in the later stage, which can effectively balance the exploration and exploitation.

Eq. (2) and (5) of the shrinking enclosing mechanism are updated to the following equations after introducing nonlinear adaptive weight \( C_1 \).

\[
\bar{X} (t + 1) = \bar{X}_p (t) - C_1 \cdot \bar{A} \cdot \left| \bar{C} \cdot \bar{X}_p (t) - \bar{X} (t) \right|
\]  

\[
\bar{X} (t + 1) = \bar{X}_{\text{rand}} (t) - C_1 \cdot \bar{A} \cdot \left| \bar{C} \cdot \bar{X}_{\text{rand}} (t) - \bar{X} (t) \right|
\]  

Eq (6) of the spiral update mechanism are updated to the following equations after introducing nonlinear adaptive weight \( C_1 \).

\[
\bar{X} (t + 1) = \bar{D} \cdot e^{\beta t} \cdot C_1 \cos (2\pi t) + \bar{X}_P (t)
\]  

From the beginning of the iteration to half of iterations, the downward trend of the nonlinear adaptive weight \( C_1 \) is slow, and the values of parameters \( \bar{A} \) and \( l \) are not decreasing significantly. The shrinking enclosing mechanism and the spiral update mechanism can enable search agents to perform global search, explore areas with excellent results in the search space, and improve the exploration capabilities of algorithm. As the number of iterations increases, the downward trend of nonlinear weight accelerates, and the values of parameters \( \bar{A} \) and \( l \) decrease significantly. The shrinking enclosing mechanism and the spiral update mechanism make the search agent move closer to the target value area, which improves the exploitation ability of the algorithm.

The introduction of nonlinear adaptive weight can adaptively adjust parameters \( \bar{A} \) and \( l \), so that the algorithm balances the exploration and exploitation stages.

**B. GOLDEN SINE OPERATOR**

The Golden Sine Algorithm (GoldenSA) is a mathematical heuristic algorithm proposed by Erkan Tanyildizi in 2017, which is inspired by the sine function in mathematics. GoldenSA uses a sine function combined with the golden section coefficient to perform an iterative search. It has good robustness and convergence speed. The sine function is represented by the abbreviation sin and the range is \([-1,1]\). The sine function is a periodic function that repeats values at regular intervals. The function period is \(2\pi\). The sine function has a special relationship with the unit circle. The relationship between the sine function and the unit circle is shown in Fig. 5. The coordinates on the sine function are relative to the y-axis coordinates of the points on the unit circle with the origin as the center and a radius of 1. Traversing the points of the sine function is equivalent to traversing all the points on the unit circle. Scanning a unit circle by a sine function is similar to exploring a search space.

In the 4th century BC, the ancient Greek mathematician Eudoxus first researched the golden section coefficient systematically and established the theory of proportion. Around 300 BC, Euclid absorbed the research results of Eudoxus when he wrote The Original Geometry, and further systematically discussed the golden section, becoming the earliest treatise on golden section. The shape designed in this proportion is very elegant, and the most harmonious dimension can be observed between the whole and the various parts. Therefore, the golden section is widely used in the field of art. The golden section coefficient is derived from the following equation:

\[
\tau = \frac{p}{q} = \frac{p + q}{p}
\]
Eq.(12) can be described as the following equations:

\[
\tau = 1 + \frac{q}{p} \tag{13}
\]

\[
\tau = 1 + \frac{1}{\tau} \tag{14}
\]

Solving Eq.(14) can gives the golden section coefficient:

\[
\tau = \frac{1 - \sqrt{5}}{2} \approx 0.618 \tag{15}
\]

The golden section coefficient does not require gradient information, only one iteration is required for each step. At the same time, the shrink step of the golden section coefficient is fixed. Therefore, combining the sine function with the golden section ratio can find the maximum or minimum value of a single unimodal function faster. The combination of the sine function and the golden section coefficient is shown in Fig. 6. The golden sine algorithm adds the golden section coefficient in the process of updating the position of the search agent, so that the algorithm can continuously reduce the search space. The algorithm searches in the area that produces excellent results instead of the entire search space, which greatly improves the convergence speed of the algorithm.

Similar to the population-based optimization algorithm, the golden sine algorithm first randomly generates initial population in the search space, that is to say:

\[
V = \text{rand} \left( N, D \right) \cdot \left( ub - lb \right) + lb \tag{16}
\]

where, \( \text{rand} \left( N, D \right) \) indicates the positions of \( N \) search agents randomly distributed in the \( D \)-dimensional space, \( ub \) is the upper bound of the search space, and \( lb \) is the upper bound of the search space. At this time, the population initialization of the golden sine algorithm has been completed.

In order to improve the random population and enable the search agent to develop towards the target value, the golden sine algorithm uses the Golden Section Search (GSS) to update the position of the search agent. In general, the flowchart of GSS method is shown in Fig. 7. The equation for GSS to update the population position is described as follows.

\[
V (i, j) = V (i, j) \cdot |\sin (r_1)| + r_2 \cdot \sin (r_2) \cdot |x_1 \cdot D (j) - x_2 \cdot V (i, j)| \tag{17}
\]

where, \( V (i, j) \) indicates the position of the search agent in the \( i \)-th dimension. \( D (j) \) indicates the position of the target value. \( r_1 \) and \( r_2 \) are random numbers, \( r_1 \in [0, 2\pi] \), \( r_2 \in [0, \pi] \). \( x_1 \) and \( x_2 \) are coefficients calculated through the golden ratio, which can drive the search agent closer to the target value. Coefficients \( x_1 \) and \( x_2 \) can be calculated by:

\[
x_1 = a \cdot \tau + b \cdot (1 - \tau) \tag{18}
\]

\[
x_2 = a \cdot (1 - \tau) + b \cdot \tau \tag{19}
\]

In the golden sine algorithm, the equations for calculating coefficients \( x_1 \) and \( x_2 \) are the same as GSS method. In order to adapt to the expression of the optimization algorithm, the values of \( a \) and \( b \) will change with the target value, which will cause the parameters \( x_1 \) and \( x_2 \) to change. Among them, the initial values of \( a \) and \( b \) are \(-\pi\) and \( \pi \). The flow chart of the parameter change with the target value is shown in Fig. 8.

The pseudo code of GoldSA is described as follows.

Initialize the population \( V \) by the Eq.(16)
Calculate the fitness of each search agent

\[ D(j) = \text{the best search agent} \]

\[ t = 1 \]

While \((t < \text{maximum number of iterations})\)

For each search agent

Update \(x_1\) and \(x_2\) by Eq. (18) (19) and Fig. 9

Update the position of the search agent by the Eq. (17)

End For

Check if any search agent goes beyond the search space and amend it

Update \(D(j)\) if there is a better solution

\[ t = t + 1 \]

End While

Return optimal value

C. IMPROVED WOA BASED ON NONLINEAR ADAPTIVE WEIGHT AND GOLDEN SINE OPERATOR

The original WOA designed the mathematical model (shrinking enclosing mechanism and the spiral update mechanism) for balanced exploitation and exploration. However, the parameters for switching exploitation and exploration are random, which will bring uncertainty to the algorithm.

Therefore, the nonlinear adaptive weight is introduced in this paper. Parameters \(\mathbf{A}\) and \(l\) can be adaptively adjusted, so that the location update mechanism can find a transition in the exploration and exploitation phases. With the introduction of nonlinear adaptive weight, the shrinking enclosing mechanism is updated to Eq. (9) and (10). The spiral update mechanism is updated to Eq. (11).

The main goal of the optimization algorithm is to find the optimal areas in the search space, and to ensure that search agents explore these areas as completely as possible. Optimization problems generally have a wide search space. Narrowing the search range will improve the algorithm’s global convergence and local exploitation. In order to achieve this process, this paper introduces an improved golden sine operator into a whale optimization algorithm. The sine function has a special relationship with the unit circle. Traversing all points on the sine function is equivalent to scanning the unit circle. The search agent can scan the search space more efficiently with a sinusoidal route, which improves the global exploration capability of the algorithm. At the same time, the golden section coefficient enables the search agent to update the distance and direction with a fixed step size, and continuously narrows the space to be explored. Search agents search in areas that return excellent results, rather than the entire search space, improving the local exploitation capabilities of the algorithm. The improved golden sine operator increases the indentation coefficient based on the golden sine operator, and enhances the moving effect of the search agent. Search agents can quickly develop into excellent regions in
order to improve the convergence speed of the algorithm. The improved golden sine operator was used to update the position of the search agent, that is to say:

\[
\vec{X}(t+1) = \vec{X}(t) \cdot |\sin(r_1)| + r_2 \cdot \sin(r_2) \cdot \left| m_1 \cdot \vec{X}_p(t) - m_2 \cdot \vec{X}(t) \right|
\]

\[m_1 = -2\pi + (1 - \tau) \cdot 2\pi\]

\[m_2 = -2\pi + \tau \cdot 2\pi\]

where, \(r_1\) and \(r_2\) are random numbers. \(r_1\) controls the moving distance of search agent. \(r_2\) controls the moving direction of search agent. \(m_1\) and \(m_2\) are improved indentation coefficients, which can drive search agents closer to the target value in longer steps.

Fig. 9 is the flowchart of NGS-WOA. The pseudo code of NGS-WOA is as follows.

Initialize the whales population \(X_i (i = 1, 2, 3 \ldots n)\)

Calculate the fitness of each search agent

\(X_p = \text{the best search agent}\)

\(t = 1\)

While (\(t < \text{maximum number of iterations}\))

For each search agent

Update \(a, A, C, l, \text{ and } p\)

If (\(p < 0.5\))

If (\(|A| < 1\))

Update the position of the search agent by the Eq.(9)

Else If (\(|A| \geq 1\))

Update the position of the search agent by the Eq.(10)

End If

Else If (\(p > 0.5\))

Update the position of the search agent by the Eq.(11)

End If

End For

Update \(r_1, r_2, m_1\) and \(m_2\)

Check if any search agent goes beyond the search space and amend it

Update \(X_p\) if there is a better solution

Update the position of the search agent by the Eq.(20)

\(t = t + 1\)

End While

Return \(X_p\)

D. TIME COMPLEXITY ANALYSIS

The time complexity of an algorithm is a function that qualitatively describes the running time of the algorithm. The time complexity is usually expressed in big \(O\) symbol. In order to calculate the time complexity, the number of operating units of the algorithm is usually estimated. Therefore, the total running time and the number of operating units of the algorithm differ by at most a constant factor. In the optimization algorithm, the time complexity is related to the number and structure of the operation units of the algorithm. For the WOA, the time complexity mainly depends on the number of search agents, the number of iterations, and the location update mechanism. The time complexity of the improved algorithm NGS-WOA mainly depends on the number of search agents, the number of iterations, and the improvement strategy. In order to evaluate the impact of the improved strategy on the algorithm running cost, the time complexity of WOA and NGS-WOA was analyzed.

The time complexity of each operation unit of WOA is described as follows.

1) The \(N\) search agents are distributed in the \(D\)-dimensional search space, which needs to run \(N \cdot D\) times.

2) Calculate the fitness value of each search agent and select the best agent as the prey, which needs to run \([N \cdot (N-1)]/2\) times.

3) Parameters \(a, A, C, l\) and \(p\) are updated respectively, which needs to run 5 times.

4) The position update operation of \(N\) search agents in the \(D\)-dimensional space, which needs to run \(N \cdot D\) times.
### TABLE 1. Test functions and specific information.

| Function                                                                 | Dim | Range        | \( f_{\text{min}} \) |
|--------------------------------------------------------------------------|-----|--------------|------------------------|
| \( F_1(x) = \sum_{i=1}^{n} x_i^2 \)                                     | 30  | \([-100,100]\) | 0                      |
| \( F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \)                         | 30  | \([-10,10]\)   | 0                      |
| \( F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{n} x_j)^2 \)                    | 30  | \([-100,100]\) | 0                      |
| \( F_4(x) = \max_i \{ |x_i|, 1 \leq i \leq n \} \)                     | 30  | \([-100,100]\) | 0                      |
| \( F_5(x) = \sum_{i=1}^{n} \left[ 100 (x_i - x_i^2)^2 + (x_i - 1)^2 \right] \) | 30  | \([-30,30]\)  | 0                      |
| \( F_6(x) = \sum_{i=1}^{n} [(x_i + 0.5)^2] \)                           | 30  | \([-10,10]\)  | 0                      |
| \( F_7(x) = \sum_{i=1}^{n} x_i^2 + \text{random}[0,1] \)                | 30  | \([-1.28,1.28]\)| 0                      |
| \( F_8(x) = \sum_{i=1}^{n} x_i^3 \sin(\sqrt{|x_i|}) \)                  | 30  | \([-500,500]\) | \(-418.9829 \times \text{Dim}\) |
| \( F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10] \)        | 30  | \([-5.12,5.12]\)| 0                      |
| \( F_{10}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \frac{x_i}{\sqrt{n}} + 1 \) | 30  | \([-600,600]\) | 0                      |
| \( F_{11}(x) = \frac{1}{n} \left[ \sin(x_1) + \sum_{i=2}^{n} (x_i - 1)^2 \left( 1 + \sin^2(3\pi x_i) \right) \right] \) + \sum_{i=1}^{n} u(x_i, 10, 100, 4) | 30  | \([-50,50]\)  | 0                      |
| \( y_1 = \frac{3 + 1}{4} u(x_i, a, k, m) = \begin{cases} \frac{k(x_i - a)^2}{x_i} & x_i > a \\ \frac{k(x_i - a)^2}{x_i} & x_i < a \\ 0 & -a < x_i < a \end{cases} \) | 30  | \([-50,50]\)  | 0                      |
| \( F_{12}(x) = 0.1 (\sin^2(3\pi x_i) + \sum_{i=1}^{n} (x_i - 1)^2 \left( 1 + \sin^2(3\pi x_i) \right) + \sum_{i=1}^{n} u(x_i, 1, 100, 4) \) | 30  | \([-50,50]\)  | 0                      |
| \( F_{13}(x) = \left( \frac{1}{500} + \sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}} \) | 2   | \([-65,65]\)  | 1                      |
| \( F_{14}(x) = \sum_{i=1}^{n} \left[ a_i - b_i \frac{x_i (b_i + b_3 x_i)}{b_i + b_3 x_i (a_i - x_i)} \right]^2 \) | 4   | \([-5,5]\)    | 0.00030                |
| \( F_{15}(x) = \left( \frac{x_1}{\pi^2 x_3} + \frac{5}{x_1} \right)^2 + 10 \left( 1 - \frac{1}{8 \pi^2} \right) \cos(x_3) + 10 \) | 2   | \([-5,5]\)    | 0.398                  |
| \( F_{16}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 \left( 19 - 14 x_1 + 3 x_2^2 + 14 x_3 + 3 x_2^2 + 14 x_4 + 3 x_4^2 + 14 x_5 + x_5^2 \right) \right] \) | 2   | \([-2,2]\)    | 3                      |
| \( \times \left[ 30 + (2 x_1 - x_2)^2 + (18 - 32 x_1 + 12 x_2^2 + 48 x_3 - 36 x_1 x_2 + 27 x_2^2) \right] \) | 3   | \([1,3]\)     | -3.86                  |
| \( F_{17}(x) = -\sum_{i=1}^{n} a_i \exp \left( -\sum_{i=1}^{n} a_i (x_i - p_i)^2 \right) \) | 6   | \([0,1]\)     | -3.32                  |
| \( F_{18}(x) = -\sum_{i=1}^{n} a_i \exp \left( -\sum_{i=1}^{n} a_i (x_i - p_i)^2 \right) \) | 4   | \([0,10]\)    | -10.1532               |
| \( F_{19}(x) = -\sum_{i=1}^{n} a_i \exp \left( -\sum_{i=1}^{n} a_i (x_i - p_i)^2 \right) \) | 4   | \([0,10]\)    | -10.4028               |
| \( F_{20}(x) = -\sum_{i=1}^{n} a_i \exp \left( -\sum_{i=1}^{n} a_i (x_i - p_i)^2 \right) \) | 4   | \([0,10]\)    | -10.5363               |
| \( F_{21}(x) = 3 x_1 + 10 \sum_{i=1}^{n} x_i^2 \)                       | 30  | \([-10,10]\)  | 0                      |
| \( F_{22}(x) = \sum_{i=1}^{n} |x_i|^4 \)                            | 30  | \([-100,100]\) | 0                      |
| \( F_{23}(x) = \sum_{i=1}^{n} x_i^2 \left( \sum_{i=1}^{n} 0.5 x_i \right)^2 + \left( \sum_{i=1}^{n} 0.5 x_i \right)^4 \) | 30  | \([-5,10]\)   | 0                      |
5) Output the optimal value (prey), which needs to run 1 times.

Each of the above operation units has undergone $T$ iterations. Therefore, the total time complexity of the WOA is $O(\text{WOA}) = T \cdot \left[ \text{ND} + \frac{(N^2 - N)}{2} + 6 \right]$.

The time complexity of each operation unit of the NGS-WOA algorithm is described as follows.

1) The $N$ search agents are distributed in the $D$-dimensional search space, which needs to run $N \cdot D$ times.

2) Calculate the fitness value of each search agent and select the best agent as the prey, which needs to run $\left\lceil \frac{N \cdot (N - 1)}{2} \right\rceil$ times.

3) Parameters $a$, $A$, $C$, $l$ and $p$ are updated respectively, which needs to run 5 times.

4) Nonlinear adaptive weight $C_1$ is updated, which needs to run 1 times.

5) The position update operation of $n$ search agents in the $d$-dimensional space, which needs to run $N \cdot D$ times.

6) Parameters $r_1$, $r_2$, $m_1$ and $m_2$ are updated respectively, which needs to run 4 times.

7) The $N$ search agents perform the golden sine operation in the $D$-dimensional space, which needs to run $N \cdot D$ times.

8) Output the optimal value (prey), which needs to run 1 times.

Each of the above operation units has undergone $T$ iterations. Therefore, the total time complexity is $O(\text{NGS-WOA}) = T \cdot \left[ \text{ND} + \frac{(N^2 - N)}{2} + 1 \right] + T \cdot \text{ND}$. The golden sine operator (GS) is used to update the optimal solution found in each iteration of the WOA algorithm.

Therefore, compared with WOA, NGS-WOA increases the time cost. In addition, GS can also be embedded in WOA as a local operator. It should be noted that although embedding the GS operator in WOA will not increase the time cost, it has a limited effect on improving the optimization performance. The influence of the introduction position of the GS operator on the optimization performance of WOA will be discussed in detail in Section B of Chapter IV.

### IV. SIMULATION EXPERIMENTS AND RESULT ANALYSIS

#### A. BENCHMARK FUNCTIONS

The experimental simulation used 28 benchmark functions to evaluate the performance of NGS-WOA. These test functions can be divided into three categories: unimodal functions, multimodal functions, and combined functions. Among them, the function $F_1$-$F_{22}$ is the CEC2005 test set. As a classic test set, you can comprehensively evaluate the performance of the algorithm. In addition, functions $F_{23}$-$F_{28}$ are CEC2017 test functions. As the latest test set, it can increase the credibility of the experiment. Functions $F_1$-$F_7$ are unimodal functions with only one global optimum. The unimodal function is used to evaluate the local exploitation ability and convergence speed of the algorithm. Functions $F_8$-$F_{13}$ are multimodal functions. Unlike unimodal functions, multimodal functions have multiple local optima. As the size of the problem increases, the number of local optima also increases. The multimodal function has important reference value for evaluating the exploration ability of the algorithm. Functions $F_{14}$-$F_{22}$ are combined functions. The difference from
multimodal functions is that they have fewer dimensions, have a small number of local optimal values, and do not allow adjustment of the dimensions. The global optimal value of the fixed-dimensional multimodal function is shifted to test
TABLE 2. Comparison of simulation performance.

| Function | GS-WOA | GS-WOA(Embed) |
|----------|--------|---------------|
|          | Ave    | Std           | Ave    | Std          |
| F1       | 0      | 0             | 6.07E-85 | 1.92E-84    |
| F2       | 0      | 0             | 8.91E-52 | 2.34E-51    |
| F3       | 0      | 0             | 2.55E-95 | 8.06E-95    |
| F4       | 0      | 0             | 3.64E-67 | 1.14E-66    |
| F5       | 1.06E-04 | 2.72E-04   | 2.66E+01 | 6.53E-01    |
| F6       | 2.35E-07 | 1.57E-07   | 0.051636 | 7.37E-02    |
| F7       | 2.00E-05 | 1.58E-05   | 1.75E-03 | 1.38E-03    |
| F8       | -12569.4615 | 5.04E-02 | -7609.3252 | 8.70E+02    |
| F9       | 0      | 0             | 6.48E+01 | 6.79E+01    |
| F10      | 8.88E-16 | 0           | 5.15E-15 | 1.50E-15    |
| F11      | 0      | 0             | 2.08E-03 | 6.59E-03    |
| F12      | 2.51E-08 | 2.53E-08   | 3.20E-03 | 1.85E-03    |
| F13      | 2.61E-07 | 3.07E-07   | 3.48E-01 | 2.65E-01    |
| F14      | 0.998   | 4.72E-11     | 1.5932  | 9.58E-01    |
| F15      | 3.25E-04 | 1.25E-05   | 8.33E-04 | 6.16E-04    |
| F16      | 0.39864 | 7.05E-04     | 0.39794 | 1.05E-04    |
| F17      | 3.0955  | 0.11414      | 3       | 5.51E-06    |
| F18      | -3.81777 | 5.21E-02   | -3.8626 | 4.42E-04    |
| F19      | -2.9547 | 1.02E-01     | -3.2713 | 6.54E-02    |
| F20      | -10.1531 | 1.44E-04  | -9.1325 | 2.149       |
| F21      | -10.4027 | 1.27E-04  | -6.4262 | 2.7839      |
| F22      | -10.5363 | 3.06E-05  | -6.0801 | 2.3817      |
| F23      | 0      | 0             | 2.94E-80 | 9.29E-80    |
| F24      | 0      | 0             | 8.29E-195 | 0          |
| F25      | 0      | 0             | 1.99E-29 | 5.79E-29    |
| F26      | 0      | 0             | 1.46E-84 | 3.61E-84    |
| F27      | 0      | 0             | 1.15E-85 | 3.53E-85    |
| F28      | 0      | 0             | 1.19E-26 | 1.41E-26    |

The position introduced by the GS operator may affect the optimization performance of the WOA algorithm. This paper proposes two methods for introducing GS operators. In the first method, the GS operator is used to optimize the optimal solution found in each iteration of WOA. In the second method, the GS operator is embedded as a local operator in WOA, replacing the spiral update mechanism. Compare the two methods to explore their effectiveness. The convergence curves of some functions are shown in Fig. 10. However, the convergence curve can only show the convergence speed and optimization accuracy of the algorithm. Therefore, the mean and variance designed in Table 2 are used to intuitively show the average accuracy and robustness. For each function, the algorithm runs independently 10 times. It can be seen from the convergence curve that the first method has better convergence speed and optimization accuracy than the second method. In addition, the first method also has obvious advantages in average accuracy and robustness.

The selection of WOA mathematical model is determined by random parameters, which increases the randomness of the algorithm. The first method updates the optimal solution found in each iteration, which can effectively avoid randomness. The advantage of the first method is that it significantly improves the performance of the algorithm, but the disadvantage is that it adds a little time complexity. The second method embeds the GS operator in WOA. However, the probability that the GS operator is selected is determined by random parameters, and the randomness of the algorithm is not eliminated. Therefore, the second method does not significantly improve the performance of the algorithm. The advantage of the second method is that it does not increase the time complexity. Although the first method adds a little time complexity, the optimization performance has been significantly improved. Therefore, this paper chooses the first method as the introduction method of GS operator.

C. COMPARISON OF NGS-WOA WITH OTHER ALGORITHMS

In order to objectively evaluate the performance of the improved WOA, this paper selects the Whale Optimization Algorithm (WOA), Gold Sine Algorithm (GoldSA), Particle Swarm Optimization (PSO) Algorithm, Firefly Algorithm (FA), Firework Algorithm (FWA), Sine Cosine Algorithm (SCA) and NGS-WOA for comparison experiments. It should be noted that NGS-WOA uses the GS operator to update the optimal solution for each iteration, so NGS-WOA has two location update evaluations in each iteration. However, other algorithms have only one location evaluation. To maintain fairness, the number of iterations of NGS-WOA is set to 500, and the number of iterations of other algorithms is set to 1000. At this time, the sum of the location update evaluation of NGS-WOA is \(2 \times 500\), which is consistent with other algorithms \((1 \times 1000)\). The parameter settings of the algorithm are listed in Table 3. The convergence curve of the algorithms are shown in Fig. 11. However, the convergence curves do not show the average accuracy and robustness of the algorithm. To further evaluate the performance of the algorithms, the algorithms were run independently 10 times for each test function. The experimental results are listed in Table 4. Three indicators listed in Table 4 are the optimal value, mean value and variance, which are used to evaluate the optimization accuracy, average accuracy, and robustness of the algorithms.

It can be seen from the convergence curves that NGS-WOA is superior to other algorithms in 50% of functions, and is the optimal algorithm in this experiment. For unimodal
functions F1-F7, the improved WOA is inferior to PSO algorithm and FA only on function F6. NGS-WOA has obvious advantages in functions F1-F5, which shows that the improved strategy proposed in this paper can effectively
(g) Convergence curves of $F_7$

(h) Convergence curves of $F_8$

(i) Convergence curves of $F_9$

(j) Convergence curves of $F_{10}$

(k) Convergence curves of $F_{11}$

(l) Convergence curves of $F_{12}$

FIGURE 11. (Continued.) Simulation experiment results.
improve the local exploitation ability and convergence speed of WOA. For multimodal functions F8-F13, NGS-WOA has obvious advantages over functions F7-F11. Especially for functions F9 and F11, the NGS-WOA and GoldSA find the
theoretical optimal value within 100 iterations and exit the iteration. The improved WOA is inferior to PSO algorithm and FA only on functions F12 and F13. It is verified that the introduction of improved golden sine operator enables
search agents to develop into excellent areas in search space. The convergence curves of the multimodal functions show that the improved algorithm can avoid the stagnation of the local optimal value and has strong global exploration ability.

FIGURE 11. (Continued.) Simulation experiment results.

TABLE 3. The specific parameter settings of the algorithm.

| Algorithm  | Main parameter settings |
|------------|-------------------------|
| NGS-WOA    | population size $n=40; T=500$ |
| GoldSA     | population size $n=40; T=1000$ |
| WOA        | population size $n=40; T=1000$ |
| PSO        | population size $n=40; T=1000$; Learning factor $c_1 = 2, c_2 = 2$; Inertia weight $W_{\text{max}} = 0.9, W_{\text{min}} = 0.1$ |
| FA         | population size $n=40; T=1000$; $\beta_0 = 1; \gamma = 1$ |
| FWA        | population size $n=40; T=1000$; |
| SCA        | population size $n=40; T=1000$ |
### TABLE 4. Comparison of simulation performance.

| Function | NGS-WOA | WOA | GoldSA | PSO | FA | FWA | SCA |
|----------|---------|-----|--------|-----|----|-----|-----|
| Best     | 0       | 6.27E-176 | 0     | 2.63E-13 | 5.40E-14 | 4.37E-55 | 8.42E-06 |
| F1 Ave   | 0       | 8.20E-169 | 0     | 5.90E-11 | 7.96E-14 | 2.46E-41 | 9.94E-03 |
| Std      | 0       | 0     | 0     | 1.47E-10 | 1.23E-14 | 7.74E-41 | 1.62E-02 |
| Best     | 0       | 3.54E-180 | 6.23E-307 | 1.30E-68 | 1.63E-11 | 8.98E-36 | 2.79E-08 |
| F2 Ave   | 0       | 8.38E-163 | 7.52E-268 | 1.50E-64 | 6.99E-11 | 8.65E-25 | 1.10E-05 |
| Std      | 0       | 2.22E-162 | 0     | 2.69E-64 | 3.57E-11 | 2.73E-24 | 1.67E-05 |
| Best     | 0       | 6.06E+03 | 0     | 3.00E+09 | 2.28E-13 | 4.77E-33 | 3.49E+02 |
| F3 Ave   | 0       | 1.84E+04 | 0     | 6.43E+07 | 6.69E+02 | 3.12E-04 | 2.31E+03 |
| Std      | 0       | 8.56E+03 | 0     | 3.72E+09 | 1.34E+01 | 9.87E-04 | 2.22E+03 |
| Best     | 0       | 1.11E-03 | 2.485E-317 | 2.09E-01 | 1.08E-07 | 2.27E-19 | 1.13E+01 |
| F4 Ave   | 0       | 2.82E+01 | 9.55E-272 | 3.28E-01 | 1.15E-07 | 5.02E-19 | 2.00E+01 |
| Std      | 0       | 2.66E+01 | 0     | 9.50E-02 | 4.82E-09 | 3.50E-19 | 6.47E+01 |
| Best     | 1.41E-09 | 2.60E+01 | 1.99E+06 | 1.77E+01 | 2.38E+01 | 2.90E+01 | 2.89E+01 |
| F5 Ave   | 1.43E+01 | 2.67E+01 | 7.05E+04 | 4.24E+01 | 2.49E+01 | 2.90E+01 | 3.94E+01 |
| Std      | 1.50E+01 | 3.00E+01 | 2.07E+03 | 2.67E+01 | 7.76E+01 | 8.60E-03 | 1.84E+01 |
| Best     | 7.80E-09 | 3.86E-03 | 2.48E-08 | 5.46E-13 | 7.41E-14 | 6.56E+7 | 3.53E+05 |
| F6 Ave   | 2.03E-07 | 9.46E-03 | 4.58E-05 | 9.81E-11 | 8.30E-14 | 1.79E+01 | 4.70E-08 |
| Std      | 2.25E-07 | 7.40E-03 | 9.89E-05 | 1.56E-10 | 7.15E-15 | 1.17E+01 | 1.24E+01 |
| Best     | 5.64E-07 | 4.23E-04 | 3.06E-06 | 95.7908 | 2.26E-02 | 1.51E-04 | 9.78E-03 |
| F7 Ave   | 5.63E-05 | 2.14E+03 | 6.26E-05 | 2.39E+02 | 2.97E-02 | 1.45E+03 | 3.02E-02 |
| Std      | 5.15E-05 | 2.26E+03 | 8.84E-05 | 1.27E+02 | 4.92E-03 | 1.53E+03 | 2.64E-02 |
| Best     | **-12569.4865** | **-12569.3488** | **-12569.4859** | **-7334.0689** | **-10431.5066** | **-3345.4754** | **-4927.4869** |
| F8 Ave   | -1.26E+04 | -1.19E+04 | -1.26E+04 | -4.75E+03 | -9.94E+03 | -1.37E+03 | -4.04E+03 |
| Std      | 1.42E-02 | 1.05E+03 | 9.53E-03 | 1.23E+03 | 4.39E+02 | 1.49E+03 | 3.30E+02 |
| Best     | 0       | 0     | 0     | 1.89E+01 | 2.79E-01 | 0     | 1.37E-04 |
| F9 Ave   | 0       | 0     | 0     | 5.45E+01 | 3.98E+01 | 14.84E2 | 1.32E+01 |
| Std      | 0       | 0     | 0     | 22.215 | 6.7968 | 15.2677 | 2.28E+01 |
| Best     | 8.88E-16 | 8.88E-16 | 8.88E-16 | 7.45E-07 | 6.15E-08 | 8.88E-16 | 1.91E-04 |
| F10 Ave  | 8.88E-16 | 3.02E-15 | 8.88E-16 | 1.14E-05 | 6.47E-08 | 14.84E2 | 1.37E+01 |
| Std      | 0       | 2.48E-15 | 0     | 9.25E-06 | 2.39E-09 | 1.53E+01 | 9.5124 |
| Best     | 0       | 0     | 0     | 2.59E-14 | 1.05E-13 | 0     | 4.40E-04 |
| F11 Ave  | 0       | 0     | 0     | 4.19E-03 | 2.71E-03 | 0     | 2.12E-01 |
| Std      | 0       | 0     | 0     | 5.45E-03 | 4.42E-03 | 0     | 1.80E-01 |
| Best     | 2.11E-09 | 3.11E-04 | 5.09E-09 | 2.99E-14 | 1.61E-16 | 8.94E-01 | 3.36E-01 |
| F12 Ave  | 1.09E-08 | 3.66E-03 | 3.25E-07 | 1.04E-02 | 2.06E-16 | 1.2753 | 6.88E-01 |
| Std      | 9.33E-09 | 5.54E-03 | 4.29E-07 | 3.28E-02 | 2.86E-17 | 0.23608 | 3.97E-01 |
| Best     | 9.20E-10 | 3.11E-03 | 3.12E-09 | 8.79E-14 | 2.33E-15 | 9.33E-01 | 4.95E-01 |
| F13 Ave  | 3.65E-07 | 6.42E-02 | 2.28E-06 | 3.30E-03 | 2.98E-15 | 1.4229 | 2.0347 |
| Std      | 7.41E-07 | 6.85E-02 | 3.15E-06 | 5.31E-03 | 4.18E-16 | 1.77E-01 | 3.1389 |
| Best     | 0.998    | 0.998  | 0.998  | 0.998  | 0.998  | 2.7838 | 2.1398 |
| F14 Ave  | 1.9745   | 2.5698 | 0.998  | 2.1846 | 0.998  | 2.8947 | 8.47E+01 |
| Std      | 3.088    | 3.027  | 2.61E-09 | 1.7852 | 1.48E-16 | 6.73E-02 | 2.54E+02 |
|   | Best  | Ave   | Std  | Std   | Ave   | Std  | Ave   | Std  | Ave   | Std  | Ave   | Std  | Ave   | Std  |
|---|-------|-------|------|-------|-------|------|-------|------|-------|------|-------|------|-------|------|
| F15 | 3.13E-04 | 3.08E-04 | 3.08E-04 | 3.07E-04 | 3.07E-04 | 6.67E-04 | 5.21E-04 |
| F16 | 3.38E-04 | 6.09E-04 | 3.41E-04 | 6.00E-04 | 5.81E-04 | 1.45E-03 | 7.88E-04 |
| F17 | 0.3979 | 0.3979 | 0.3979 | 0.3978 | 0.3978 | 0.40007 | 0.39813 |
| F18 | 0.39834 | 0.39879 | 0.39873 | 0.39879 | 0.39879 | 0.41362 | 0.39868 |
| F19 | 5.54E-04 | 2.28E-07 | 2.07E-03 | 0.0 | 0.163E-02 | 3.96E-04 |
| F20 | 3.0003 | 3.0004 | 3.0004 | 3.0004 | 3.0008 | 3.0007 | 3.0007 |
| F21 | 5.86E-02 | 8.03E-06 | 1.39E+01 | 9.00E-16 | 8.24E-16 | 8.72E-03 | 1.51E-05 |
| F22 | -3.8617 | -3.8628 | -3.8627 | -3.8628 | -3.8628 | -3.8624 | -3.8548 |
| F23 | -3.7192 | -3.7966 | -3.7864 | -3.7864 | -3.7864 | -3.7864 | -3.7864 |
| F24 | 1.26E-01 | 3.19E-03 | 9.24E-02 | 3.81E-03 | 4.44E-16 | 9.90E-04 | 4.07E-04 |
| F25 | -3.1946 | -3.3219 | -3.2013 | -3.322 | -3.322 | -3.1758 | -3.1981 |
| F26 | -2.9294 | -3.2944 | -2.9677 | -3.2313 | -3.2863 | -3.1273 | -2.9925 |
| F27 | 1.94E-01 | 5.76E-02 | 4.31E-01 | 8.64E-02 | 5.74E-02 | 3.65E-02 | 2.74E-01 |
| F28 | -10.1532 | -10.1531 | -10.1532 | -10.1532 | -10.1532 | -10.1532 | -10.1532 |
| F29 | -10.1531 | -9.6427 | -10.1524 | -7.8904 | -10.1532 | -3.5819 | -2.2504 |
| F30 | 1.95E-04 | 1.6119 | 1.47E-03 | 3.0034 | 1.87E-15 | 7.46E-01 | 2.0658 |
| F31 | -10.4028 | -10.4029 | -10.4028 | -10.4029 | -10.4029 | -6.9583 | -6.328 |
| F32 | 3.17E-05 | 2.5486 | 1.36E-04 | 2.5415 | 2.37E-15 | 1.2861 | 8.71E-01 |
| F33 | -10.5363 | -10.5364 | -10.5364 | -10.5364 | -10.5364 | -4.1393 | -6.6962 |
| F34 | 4.23E-05 | 2.8411 | 7.75E-04 | 4.15E-15 | 1.87E-15 | 0.60007 | 1.4862 |
| F35 | 0 | 1.10E-175 | 0 | 1.35E+06 | 4.6438 | 2.80E-51 | 7.73E-02 |
| F36 | 0 | 1.70E+160 | 0 | 3.00E+01 | 4.55E+01 | 7.12E-41 | 2.59E+01 |
| F37 | 0 | 4.52E+160 | 0 | 4.83E+01 | 2.69E+01 | 2.21E-40 | 4.18E+01 |
| F38 | 0 | 1.99E+267 | 0 | 1.69E+19 | 1.85E+06 | 3.65E+127 | 6.26E-11 |
| F39 | 0 | 4.94E-231 | 0 | 1.10E-13 | 3.3528E+11 | 1.40E-92 | 7.16E+04 |
| F40 | 0 | 1.0E+00 | 0 | 3.47E-13 | 7.56E+11 | 4.36E-92 | 2.25E+05 |
| F41 | 0 | 2.57E+02 | 0 | 4.3298 | 8.14E-16 | 3.86E-20 | 1.3994 |
| F42 | 0 | 4.62E+02 | 0 | 1.63E+02 | 1.25E-15 | 5.95E-01 | 8.1312 |
| F43 | 0 | 1.20E+02 | 0 | 1.64E+02 | 2.39E-16 | 1.8815 | 5.3155 |
| F44 | 0 | 1.70E-172 | 0 | 3.45E-09 | 7.93E+05 | 1.50E-61 | 1.36E-04 |
| F45 | 0 | 8.50E-161 | 0 | 2.89E-06 | 1.42E+06 | 4.04E-44 | 2.96E-01 |
| F46 | 0 | 1.94E-160 | 0 | 4.37E-06 | 4.51E+05 | 1.28E-43 | 8.79E-01 |
| F47 | 0 | 9.57E-176 | 0 | 7.05E-13 | 3.14E+04 | 1.17E-61 | 3.03E-07 |
| F48 | 0 | 3.64E-160 | 0 | 6.70E-11 | 3.77E+04 | 3.14E-45 | 0.0069622 |
| F49 | 0 | 1.15E-159 | 0 | 1.33E-10 | 3.50E+03 | 9.31E-45 | 0.021215 |
| F50 | 0 | 1.27E-70 | 0 | 1.1279 | 1.3593 | 2.89E-16 | 2.35E-02 |
| F51 | 0 | 2.53E-64 | 0 | 1.3681 | 1.5038 | 9.24E-14 | 1.21E-01 |
| F52 | 0 | 7.31E-64 | 0 | 1.66E-01 | 8.09E-02 | 1.24E-13 | 7.99E-02 |
TABLE 5. P-value test of Wilcoxon rank.

| Function | NGS-WOA vs WOA | NGS-WOA vs GoldSA | NGS-WOA vs PSO | NGS-WOA vs FA | NGS-WOA vs FWA | NGS-WOA vs SCA |
|----------|----------------|-------------------|----------------|--------------|---------------|----------------|
| F1       | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F2       | 6.39E-05       | 7.51E-04          | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F3       | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F4       | 6.39E-05       | 6.39E-05          | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F5       | 1.41E-01       | 9.70E-01          | 2.20E-03       | 1.41E-01     | 1.83E-04     | 4.40E-04       |
| F6       | 1.83E-04       | 2.80E-03          | 1.83E-04       | 1.83E-04     | 1.83E-04     | 1.83E-04       |
| F7       | 1.83E-04       | 2.41E-01          | 1.83E-04       | 1.83E-04     | 1.83E-04     | 1.83E-04       |
| F8       | 1.38E-04       | 2.41E-01          | 1.38E-04       | 1.38E-04     | 1.38E-04     | 1.58E-04       |
| F9       | NaN            | NaN               | 6.39E-05       | 6.39E-05     | NaN          | 6.39E-05       |
| F10      | 5.40E-03       | NaN               | 6.39E-05       | 6.39E-05     | NaN          | 6.39E-05       |
| F11      | 1.68E-01       | NaN               | 6.39E-05       | 6.39E-05     | NaN          | 6.39E-05       |
| F12      | 1.83E-04       | 9.10E-03          | 1.38E-04       | 1.38E-04     | 1.38E-04     | 1.38E-04       |
| F13      | 1.83E-04       | 6.40E-02          | 2.80E-03       | 1.83E-04     | 1.83E-04     | 1.83E-04       |
| F14      | 4.60E-03       | 2.11E-02          | 1.82E-01       | 6.34E-05     | 4.73E-01     | 4.73E-01       |
| F15      | 1.13E-02       | 2.12E-01          | 2.80E-03       | 1.83E-04     | 1.30E-03     | 1.30E-03       |
| F16      | 1.83E-04       | 2.46E-04          | 1.70E-03       | 6.39E-05     | 1.14E-01     | 1.13E-02       |
| F17      | 1.83E-04       | 5.71E-01          | 6.39E-05       | 6.39E-05     | 1.40E-02     | 1.38E-04       |
| F18      | 2.46E-04       | 7.57E-02          | 6.39E-05       | 6.39E-05     | 1.00E-03     | 1.70E-03       |
| F19      | 2.46E-04       | 6.40E-02          | 2.16E-04       | 1.54E-04     | 3.76E-02     | 0.3075         |
| F20      | 1.40E-01       | 9.10E-01          | 4.69E-01       | 6.39E-05     | 1.83E-04     | 1.83E-04       |
| F21      | 2.20E-03       | 1.86E-01          | 1.32E-01       | 6.39E-05     | 1.83E-04     | 1.83E-04       |
| F22      | 7.69E-04       | 0.5205            | 0.0017         | 6.39E-05     | 1.83E-04     | 1.83E-04       |
| F23      | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F24      | 1.83E-04       | 6.39E-05          | 1.83E-04       | 1.83E-04     | 1.83E-04     | 1.83E-04       |
| F25      | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F26      | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F27      | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |
| F28      | 6.39E-05       | NaN               | 6.39E-05       | 6.39E-05     | 6.39E-05     | 6.39E-05       |

The compound functions F14–F22 has a small dimension and is mainly used to evaluate the optimization accuracy of the algorithm. Therefore, the difference in algorithm convergence speed is not obvious. For functions F23–F28, the improved algorithm has obvious advantages. The convergence speed of NGS-WOA is worse than that of GoldSA only in function F3. As can be seen from Table 4, NGS-WOA found a theoretical optimal value on 54% of functions. WOA found a theoretical optimal value on 18% of functions. GoldSA found a theoretical optimal value on 46% of functions. PSO algorithm, FA, and FWA found theoretical optimal values on 7% of functions. SCA found a theoretical optimal value on 3% of functions. The improved WOA has the best optimization accuracy in this experiments. NGS-WOA also performs well in terms of average accuracy and robustness. NGS-WOA is more competitive than other algorithms, which shows that the improved WOA has better optimization ability, and the excellent robustness makes the algorithm less susceptible to randomness.

In order to objectively show the comparison between NGS-WOA and other algorithms, this paper uses Wilcoxon rank sum test [51] to further analyze the experimental results. As a non-parametric statistical test, Wilcoxon rank sum test uses sample rank instead of sample value test, which can analyze whether the difference between two samples is significant. It can also test whether the distribution
functions of two populations are equal. Table 5 describes the p-value test of wilcoxon rank sum test. The statistical results determined the level of difference between NGS-WOA and other algorithms, which can be recorded as the p-value.
If $p$ exceeds 0.05, it can be understood that there is no significant change between the two samples. If $p$ is less than 0.05 and close to 0, it means that there is a significant difference between the two samples. If the $p$-value result is NaN, there is no difference between the two samples. The test results listed in Table 5 show that the $p$-value of NGS-WOA in most functions is less than 0.05. Wilcoxon rank sum test further shows that NGS-WOA has significant advantages over other algorithms.

In summary, the results in this section show that NGS-WOA is the best algorithm, the second is the GoldSA algorithm, and the third is the WOA algorithm. NGS-WOA is not only better than WOA, but also better than GoldSA, which verifies the effectiveness of the improved strategy proposed in this paper. Nonlinear weight enables the position update mechanism of whale algorithm to adaptively balance the exploration and exploitation phases. In addition, the introduction of improved golden sine operator enhances the moving effect of search agents, efficiently searches only in areas that produce good results, which improves the exploitation and exploration capabilities of algorithm.

### D. COMPARISON OF NG-SWOA AND OTHER IMPROVED ALGORITHMS.

Section C has compared NGS-WOA with some algorithms. The experimental results show that NGS-WOA has

| Function | NGS-WOA | LWOA | CPWOA | PSOGSA | VPSO |
|----------|--------|------|-------|--------|------|
| Ave | Std | Ave | Std | Ave | Std | Ave | Std | Ave | Std |
| F1 | 0 | 0 | 0 | 0 | 2.86E-121 | 9.04E-121 | 3.00E+03 | 6.75E+03 | 1.81E-154 | 3.35E-154 |
| F2 | 0 | 0 | 3.51E-322 | 0 | 5.21E-163 | 0 | 1.20E+01 | 3.45E+01 | 9.45E-80 | 2.99E-79 |
| F3 | 0 | 0 | 0 | 0 | 1.59E-03 | 3.36E-03 | 9.22E-22 | 4.75E-22 | 1.22E-171 | 0 |
| F4 | 0 | 0 | 0 | 0 | 9.06E-03 | 2.13E-02 | 1.99E-11 | 4.97E-12 | 2.03E-84 | 6.40E-84 |
| F5 | 1.43E+01 | 1.50E+01 | 2.82E+01 | 5.14E-01 | 1.42E+01 | 1.49E+01 | 4.81E+01 | 7.14E+01 | 2.89E-01 | 8.76E-02 |
| F6 | 2.03E-07 | 2.25E-07 | 5.80E+00 | 2.16E-01 | 4.96E-03 | 1.18E-03 | 9.90E+02 | 3.13E+03 | 6.60E7 | 3.28E-01 |
| F7 | 5.63E-05 | 5.15E-05 | 1.02E-04 | 1.22E-04 | 2.32E-03 | 3.25E-03 | 6.42E-02 | 2.96E-02 | 2.18E-04 | 1.55E-04 |
| F8 | -1.26E+04 | 1.42E-02 | -8.55E+05 | 7.52E+05 | -1.26E+04 | 4.79E-02 | -7.77E+03 | 9.53E+02 | -3.10E+03 | 2.88E+02 |
| F9 | 0 | 0 | 0 | 0 | 0 | 1.24E+02 | 3.51E+01 | 0 | 0 |
| F10 | 8.88E-16 | 0 | 1.24E-15 | 1.12E-15 | 2.66E-15 | 2.51E-15 | 1.47E+01 | 5.1026 | 8.88E-16 | 0 |
| F11 | 1.09E-08 | 9.33E-09 | 7.91E-01 | 5.91E-02 | 3.31E-04 | 2.25E-04 | 3.7727 | 4.0523 | 1.203 | 2.35E-01 |
| F12 | 3.65E-07 | 7.41E-07 | 2.6573 | 6.76E-02 | 1.40E-02 | 1.95E-02 | 4.10E+07 | 1.30E+08 | 2.8558 | 6.38E-02 |
| F13 | 1.9745 | 3.088 | 2.98E+00 | 6.16E-10 | 9.98E-01 | 9.59E-08 | 3.3381 | 4.59E+00 | 1.0714 | 1.75E-01 |
| F14 | 3.88E-04 | 2.84E-05 | 6.85E-04 | 4.36E-04 | 7.70E-04 | 3.83E-04 | 1.24E-02 | 1.03E-02 | 1.08E-03 | 3.55E-04 |
| F15 | 3.98E-01 | 5.54E-04 | 4.02E-01 | 4.46E-03 | 3.98E-01 | 3.10E-12 | 3.98E-01 | 0 | 4.01E-01 | 2.23E-03 |
| F16 | 3.0452 | 5.86E-02 | 3.00E+00 | 3.57E-05 | 3.00E+00 | 1.12E-03 | 1.11E+01 | 2.56E+01 | 3.00E+00 | 7.33E-04 |
| F17 | -3.7192 | 1.26E-01 | -3.8453 | 1.14E-02 | -3.8619 | 2.37E-03 | -3.8628 | 7.83E-16 | -3.8611 | 1.01E-03 |
| F18 | -2.9294 | 1.94E-01 | -2.7657 | 2.40E-01 | -3.2325 | 1.06E-01 | -3.2441 | 6.99E-02 | -3.1652 | 4.66E-02 |
| F19 | -1.0153 | 1.95E-04 | -1.7067 | 1.4788 | -9.643 | 1.612 | -5.8866 | 3.126 | -4.0211 | 4.65E-01 |
| F20 | -1.4028 | 3.17E-05 | -1.9075 | 1.3547 | -9.8688 | 1.6888 | -4.4277 | 2.3778 | -3.77 | 5.95E-01 |
| F21 | -1.5363 | 4.23E-05 | -1.9406 | 1.3315 | -10.5363 | 2.39E-04 | -6.7488 | 4.0077 | -4.128 | 5.40E-01 |
| F22 | 0 | 0 | 0 | 0 | 5.60E-167 | 0 | 9.97E-05 | 1.60E-04 | 9.98E-147 | 2.26E-146 |
| F23 | 0 | 0 | 0 | 0 | 5.27E-126 | 1.67E-125 | 1.01E+43 | 3.16E+43 | 5.98E-269 | 0 |
| F24 | 0 | 0 | 5.61E-315 | 0 | 3.30E+02 | 8.67E+01 | 1.16E-04 | 2.35E-04 | 1.87E-45 | 5.89E-45 |
| F25 | 0 | 0 | 0 | 0 | 1.79E+128 | 5.65E-138 | 3.48E+07 | 3.51E+07 | 3.54E-149 | 9.56E-149 |
| F26 | 0 | 0 | 0 | 0 | 4.04E-122 | 1.28E-121 | 1.64E+04 | 1.32E+04 | 7.29E-144 | 2.30E-143 |
| F27 | 0 | 0 | 0 | 0 | 0 | 0 | 2.3445 | 1.09E-01 | 8.08E-39 | 2.47E-38 |
obvious advantages. In order to further explore the performance of NGS-WOA, this paper selects other improved algorithms for comparative experiments, such as IWOA [52], CPWOA [53], PSOGSA [54], and VPSO [55]. Choose the
function F1-F28 to experiment. The convergence curves of some functions are shown in Fig. 12. For each function, the algorithm runs independently 10 times. The statistical results are shown in Table 6. It can be seen from the experimental results that NGS-WOA has the best optimization ability. NGS-WOA has obvious advantages in
FIGURE 13. (Continued.) Simulation experiment results.
convergence speed, average accuracy and robustness. LWOA is the second-best algorithm, with good accuracy in many functions. CP-WOA is the third best algorithm. In summary, the comparison experiments with other improved algorithms further verify that NGS-WOA has strong performance.

**E. EFFECTIVENESS ANALYSIS OF IMPROVED STRATEGIES**

This paper proposes two improvements based on the WOA: Firstly, the nonlinear adaptive weight is introduced to enable the algorithm to balance exploitation and exploration. Secondly, in order to further increase the exploitation and exploration capabilities of the algorithm, an improved golden sine operator is introduced. Section B has preliminary verified that the improved strategy can effectively improve the performance of the whale optimization algorithm. In order to further evaluate the effect of the improved strategy on the performance of the algorithm, the improved WOA (NGS-WOA),

[FIGURE 13. (Continued.) Simulation experiment results.]

whale optimization algorithm (WOA), and WOA only based on golden sine operator (GS-WOA) were selected for comparison experiments. By using functions F1-F22, and the convergence curves are shown in Fig. 13. Each function runs independently 10 times. The experimental results are listed in Table 7. It can be seen from the convergence curves that whether for unimodal functions F1-F7 or multimodal functions F8-F13, NGS-WOA has better convergence speed and optimization accuracy than WOA, which indicates that the two improvement strategies proposed in this paper can effectively improve performance of algorithm. Because the dimensions of functions F14-F22 are not large, the differences between algorithms are not obvious. WOA based on the golden sine operator (GS-WOA) is significantly better than WOA, which shows that improved sine operator can make the search agent scan the space efficiently and improve the exploration ability of the algorithm. The increase
### TABLE 7. Comparison of improvement strategies.

| Function | NGS-WOA | WOA | GS-WOA |
|----------|---------|-----|--------|
| Best     | 0       | 6.27E-176 | 0 |
| Ave      | 0       | 8.20E-169 | 0 |
| Std      | 0       | 0       | 0 |
| F1       | 0       | 3.54E-180 | 0 |
| Best     | 0       | 6.06E+03 | 0 |
| Ave      | 0       | 6.83E-163 | 0 |
| Std      | 0       | 2.22E-162 | 0 |
| F2       | 0       | 8.56E+03 | 0 |
| Best     | 0       | 1.11E-03 | 0 |
| Ave      | 0       | 1.84E+04 | 0 |
| Std      | 0       | 2.66E+01 | 0 |
| F3       | 0       | 2.60E+01 | 4.87E-06 |
| Best     | 1.41E-09 | 5.65E-05 | 0 |
| Ave      | 1.43E+01 | 2.67E+01 | 5.65E-05 |
| Std      | 1.50E+01 | 3.00E-01 | 4.11E-05 |
| F4       | 0       | 3.86E-03 | 4.88E-09 |
| Best     | 7.80E-09 | 9.46E-03 | 3.03E-07 |
| Ave      | 2.03E-07 | 7.40E-03 | 2.56E-07 |
| Std      | 2.25E-07 | 4.23E-04 | 1.50E-06 |
| F5       | 0       | 2.14E-03 | 3.68E-05 |
| Best     | 5.63E-05 | 2.26E-03 | 4.74E-05 |
| Ave      | 5.15E-05 | 2.65E-03 | 1.2569.4865 |
| Std      | 5.64E-07 | 4.23E-04 | 1.50E-06 |
| F6       | 0       | 2.56E-07 | 2.56E-07 |
| Best     | 2.26E-03 | 4.74E-05 | -12569.4865 |
| Ave      | 1.26E+04 | -1.19E+04 | -1.26E+04 |
| Std      | 1.42E-02 | 1.05E+03 | 9.98E-03 |
| F7       | 0       | 0       | 0 |
| Best     | 8.88E-16 | 8.88E-16 | 8.88E-16 |
| Ave      | 8.88E-16 | 3.02E-15 | 8.88E-16 |
| Std      | 0       | 2.48E-15 | 0 |
| F8       | 0       | 0       | 0 |
| Best     | 8.88E-16 | 8.88E-16 | 8.88E-16 |
| Ave      | 8.88E-16 | 3.02E-15 | 8.88E-16 |
| Std      | 0       | 2.48E-15 | 0 |
| F9       | 0       | 0       | 0 |
| Best     | 2.11E-09 | 3.11E-04 | 2.55E-11 |
| Ave      | 1.09E-08 | 3.66E-03 | 2.41E-08 |
| Std      | 9.33E-09 | 5.54E-03 | 2.28E-08 |
| F10      | 0       | 3.11E-03 | 1.86E-09 |
| Best     | 9.20E-10 | 3.11E-03 | 1.86E-09 |
| Ave      | 3.65E-07 | 6.42E-02 | 3.01E-07 |
| Std      | 3.41E-07 | 6.85E-02 | 4.82E-07 |
| F11      | 0       | 0       | 0 |
| Best     | 2.11E-09 | 3.11E-04 | 2.55E-11 |
| Ave      | 1.09E-08 | 3.66E-03 | 2.41E-08 |
| Std      | 9.33E-09 | 5.54E-03 | 2.28E-08 |
| F12      | 0       | 3.11E-03 | 1.86E-09 |
| Best     | 9.20E-10 | 3.11E-03 | 1.86E-09 |
| Ave      | 3.65E-07 | 6.42E-02 | 3.01E-07 |
| Std      | 7.41E-07 | 6.85E-02 | 4.82E-07 |
| F13      | 0       | 0       | 0 |
| Best     | 0.998   | 0.998   | 0.998 |
| Ave      | 1.9745  | 2.5698  | 1.9745 |
| Std      | 3.088   | 3.027   | 3.088 |
| F14      | 0       | 3.13E-04 | 3.08E-04 |
| Best     | 3.13E-04 | 3.08E-04 | 3.10E-04 |
of the indentation coefficient enhances the moving effect of the search agent. The search agent explores excellent areas and improves the exploitation of the algorithm. NGS-WOA has better optimization effect than GS-WOA, indicating that the introduction of nonlinear adaptive weights can further balance exploitation and exploration. For each function, the results of 10 independent runs of the algorithm are selected for Wilcoxon rank sum test. Table 8 describes the p-value test for Wilcoxon rank sum test. It can be seen from the p-value test that NGS-WOA has significant advantages over WOA and GS-WOA, which indicates that a variety of improvement strategies can significantly improve the performance of the algorithm.

F. HIGH-DIMENSIONAL FUNCTION OPTIMIZATION

The search space for practical optimization problems is complex, with high dimensions and noise. For example, different data sets cause neural networks to have different structures. When the optimization algorithm is used as a neural network trainer, the dimensions that need to be optimized may be more than 50 dimensions [52]. In order to verify the effectiveness of the algorithm for solving high-dimensional optimization problems, this paper applies NGS-WOA to experiments of high-dimensional function optimization. The dimensions of the fixed dimension functions \( F_{14} - F_{22} \) must not be changed. Therefore, the dimensions \( D = 30 \) of functions \( F_1 - F_{13} \) are increased to \( D = 50 \), \( D = 100 \), \( D = 150 \), and \( D = 200 \), respectively. The experimental results are listed in Table 9. Generally, benchmark functions create the search space based on dimensions, number of variables, and constraints. The calculation cost of the algorithm will increase with the increase of the dimension, and the algorithm is prone to fall into a dimensional disaster. The unimodal functions \( F_1 - F_7 \) will increase the search range as the dimension increases, and the exploitation capability of the search agent will decrease. In addition, an increase in the dimension will increase the local optimal value of the multimodal functions \( F_8 - F_{13} \), which may cause the stagnation of local optima. If the exploration capability of algorithm is weak, the search agent cannot jump out of the local optimum. Therefore, there are challenges in high-dimensional optimization problems. It can be seen from the experimental
results that NGS-WOA found the theoretical optimal value of the 54% functions in four high-dimensional situations. Among them, for the unimodal functions \(F_1 - F_7\), the increase of the dimensions has no obvious effect on the improved WOA, which indicates that NGS-WOA has strong exploitation ability. For multimodal functions \(F_8 - F_{13}\), the average accuracy and robustness of NGS-WOA is excellent, without falling into dimensional disaster. It shows that the algorithm has strong exploration ability. It should be noted that the optimal value of the function \(F_8\) is \(f_{\text{min}} = -418.9829 \cdot D\), the global optimal value will shift with the dimensions, and NGS-WOA can still find the theoretical optimal value. To verify the performance difference of the improved WOA in different dimensions, Table 10 describes the p-value test for Wilcoxon rank sum test. The p-value test results show that there is no difference in performance between high-dimensional optimization and low-dimensional optimization in most functions. Wilcoxon rank sum test further illustrates that the improved WOA has strong optimization ability and robustness, and can avoid stagnation of local optimal values. NGS-WOA can effectively solve high-dimensional optimization problems. In order to explore the impact of variable population and iteration times on high-dimensional optimization performance, Table 11 shows the statistical results of different variables in 100-dimensions. The experimental results show that increasing the variable population and the number of iterations has little effect on NGS-WOA.

**G. ENGINEERING OPTIMIZATION DESIGN**

In order to verify the effectiveness of the improved WOA for engineering optimization, NGS-WOA was applied to the optimization design of welded beams. The purpose of welding beam optimization is to minimize the manufacturing cost of welding beams. The four optimization constraints are shear stress \(\tau\), bending stress in the beam \(\theta\), buckling load \(p_c\), and deflection of the beam \(\delta\). These four variables that need to be optimized are the thickness of the weld seam \(h\), the length of the clamp beam \(l\), the height of the bar \(r\), and the thickness of the bar \(b\). The objective function, optimization constraints, and optimization variables of the welding beam optimization
### TABLE 9. High-dimensional optimization results.

| Function | Performance | NGS-WOA |
|----------|-------------|---------|
|          |             | D=50    | D=100   | D=150   | D=200   |
|          | Best        | 0       | 0       | 0       | 0       |
| F1       | Ave         | 0       | 0       | 0       | 0       |
| F2       | Std         | 0       | 0       | 0       | 0       |
|          | Best        | 0       | 0       | 0       | 0       |
| F3       | Ave         | 0       | 0       | 0       | 0       |
| F4       | Std         | 0       | 0       | 0       | 0       |
|          | Best        | 0       | 0       | 0       | 0       |
| F5       | Ave         | 4.85E+00| 2.20E-04| 4.65E-04| 4.73E-04|
| Std      | 1.53E+04   | 2.18E-04| 5.46E-04| 9.91E-04|
|          | Best        | 5.06E-09| 1.05E-08| 2.41E-09| 1.67E-07|
| F6       | Ave         | 3.60E-07| 2.66E-07| 4.55E-07| 1.09E-06|
| Std      | 6.58E-07   | 4.64E-07| 6.46E-07| 1.13E-06|
|          | Best        | 5.44E-06| 1.35E-06| 4.04E-06| 1.03E-06|
| F7       | Ave         | 2.53E-05| 2.26E-05| 2.48E-05| 1.95E-05|
| Std      | 1.92E-05   | 2.29E-05| 2.17E-05| 1.51E-05|
|          | Best        | -20949.1442| -41898.2887| -62847.4328| -83796.5771|
| F8       | Ave         | -2.09E+04| -4.19E+04| -6.28E+04| -8.38E+04|
| Std      | 3.43E-02   | 9.53E-02| 4.24E-02| 1.11E-01|
|          | Best        | 0       | 0       | 0       | 0       |
| F9       | Ave         | 0       | 0       | 0       | 0       |
| Std      | 0          | 0       | 0       | 0       |
|          | Best        | 8.88E-16| 8.88E-16| 8.88E-16| 8.88E-16|
| F10      | Ave         | 8.88E-16| 8.88E-16| 8.88E-16| 8.88E-16|
| Std      | 0          | 0       | 0       | 0       |
|          | Best        | 0       | 0       | 0       | 0       |
| F11      | Ave         | 0       | 0       | 0       | 0       |
| Std      | 0          | 0       | 0       | 0       |
|          | Best        | 3.16E-11| 5.82E-11| 3.11E-10| 3.17E-10|
| F12      | Ave         | 3.39E-09| 1.74E-08| 1.24E-09| 1.58E-08|
| Std      | 4.89E-09   | 4.25E-08| 9.36E-10| 2.75E-08|
|          | Best        | 2.19E-09| 4.97E-09| 5.83E-09| 2.02E-08|
| F13      | Ave         | 9.73E-07| 5.28E-07| 9.91E-07| 1.10E-06|
| Std      | 1.74E-06   | 6.74E-07| 1.09E-06| 1.80E-06|
problem are expressed by the following equations.

Consider \( \vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b] \)

Minimize \( f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4 (14.0 + x_2) \)

\[ \begin{align*}
  &\text{Subject } g_1(\vec{x}) = t(\vec{x}) - \tau_{\max} \leq 0 \\
  &g_2(\vec{x}) = \theta(\vec{x}) - \theta_{\max} \leq 0 \\
  &g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0
\end{align*} \]
TABLE 12. Comparison of optimization costs.

| Algorithm   | Optimum variables | Optimum cost |
|-------------|-------------------|--------------|
| NGS-WOA    | 0.20722 3.393969 9.018874 0.207225 | 1.725        |
| WOA        | 0.20539 3.484293 9.037426 0.206276 | 1.730499     |
| GSA        | 0.18212 3.856979 10 0.202376 | 1.879952     |
| GA (Deb)   | 0.2489 6.173 8.1789 0.2533 | 2.4331       |
| HS (Lee and Geem) | 0.2442 6.2231 8.2915 0.2443 | 2.3807       |
| Random     | 0.4575 4.7313 5.0853 0.66 | 4.1185       |
| Simplex    | 0.2792 5.6256 7.7512 0.2796 | 2.5507       |
| David      | 0.2434 6.2552 8.2915 0.2444 | 2.3841       |
| APPROX     | 0.2444 6.2189 8.2915 0.2444 | 2.3815       |

TABLE 13. Comparison of statistical results.

| Algorithm   | Average | Standard Deviation |
|-------------|---------|--------------------|
| NGS-WOA    | 2.0632  | 0.3721             |
| WOA        | 1.732   | 0.0226             |
| PSO        | 1.7422  | 0.01275            |
| GSA        | 3.5761  | 1.2874             |

\[
g_4(\bar{x}) = x_1 - x_4 \leq 0
\]
\[
g_5(\bar{x}) = p - p_c(\bar{x}) \leq 0
\]
\[
g_6(\bar{x}) = 0.125 - x_1 \leq 0
\]
\[
g_7(\bar{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) \leq 0
\]

Variable range \[0.1 \leq x_1 \leq 2\]
\[0.1 \leq x_2 \leq 10\]
\[0.1 \leq x_3 \leq 10\]
\[0.1 \leq x_4 \leq 2\] (23)

In order to objectively demonstrate the performance of NGS-WOA, WOA and GSA in [43] were selected. In addition, mathematical techniques (constrained correction of fixed costs [57] and penalty functions [58]) or meta-heuristic algorithm techniques such as PSO algorithm [59], Evolutionary Strategy (ES) [60], GA [61], improved harmony search (HS) [62], Differential Evolution (DE) [63] and Ray Optimization (RO) algorithm [64] were selected. The optimization results listed in Table 14 show that the improved WOA has the best optimization cost. The comparison of the statistical results in Table 15 shows that the improved WOA is better than GSA, PSO algorithm and WOA in terms of mean and variance.

H. DISCUSSION OF NGS-WOA ALGORITHM

In view of the shortcomings of the WOA algorithm, this article will be improved from two aspects. First, the nonlinear weights can make the parameters adaptively change with iteration, eliminating the uncertainty caused by random parameters. Second, the golden sine operator optimizes the optimal solution found in each iteration of WOA, which makes up for the shortcomings of the mathematical model determined by the random parameters. In addition, the golden sine operator updates the optimal solution found in each iteration of WOA, which further improves the optimization performance of the algorithm. Compared with other algorithms, NGS-WOA has better optimization performance, and has the advantages of high precision and high convergence speed. Compared with the original WOA, the disadvantage of NGS-WOA is that it adds a little time complexity. However, the optimization performance has been significantly improved. Experimental results show that NGWSWOA can effectively solve high-dimensional optimization and engineering optimization problems, which lays theoretical foundation for the application of NGS-WOA.

Consider \[\bar{x} = [x_1, x_2, x_3] = [d, D, n]\]

Minimize \[f(\bar{x}) = (2 + x_3)x_1^2x_2\]

Subject \[g_1(\bar{x}) = 1 - \frac{x_2^3x_3}{717851}\] \[\leq 0\]

\[g_2(\bar{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_3^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0\]

\[g_3(\bar{x}) = 1 - \frac{140.45x_1}{x_2x_3} \leq 0\]

\[g_4(\bar{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0\]

Variable range \[0.05 \leq x_1 \leq 2\]
\[0.25 \leq x_2 \leq 1.3\]
\[2 \leq x_3 \leq 15\] (24)

The purpose of spring optimization is to minimize the weight of the spring. There are three design variables: wire diameter \(d\), average coil diameter \(D\), and number of active coils \(n\). The optimal design of the spring is expressed by the following equation:

\[g_4(\bar{x}) = x_1 - x_4 \leq 0\]

\[g_5(\bar{x}) = p - p_c(\bar{x}) \leq 0\]

\[g_6(\bar{x}) = 0.125 - x_1 \leq 0\]

\[g_7(\bar{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) \leq 0\]
TABLE 14. Comparison of optimization costs.

| Algorithm          | Optimum variables | Optimum cost |
|--------------------|-------------------|--------------|
| NGS-WOA            | 0.061381          | 0.01266      |
| WOA                | 0.051207          | 0.1200403    |
| GSA                | 0.050276          | 0.0127022    |
| PSO (Ha and Wang)  | 0.051728          | 0.0126747    |
| ES (Coello and Montes) | 0.051989     | 0.012681     |
| GA (Coello)        | 0.05148           | 0.0127048    |
| RO (Kaveh and Khayatzad) | 0.05137        | 0.0126788    |
| Improved HS (Mahdavi et al.) | 0.051154    | 0.0126706    |
| DE (Huang et al.)  | 0.051609          | 0.0126702    |
| Mathematical optimization(Belegundu) | 0.053396       | 0.0127303    |
| Constraint correction(Arora) | 0.05         | 0.3159       |

TABLE 15. Comparison of statistical results.

| Algorithm             | Average  | Standard Deviation |
|-----------------------|----------|--------------------|
| NGS-WOA               | 0.012661 | 4.88E-07           |
| WOA                   | 0.0127   | 0.0003             |
| PSO                   | 0.0139   | 0.0033             |
| GSA                   | 0.0136   | 0.0026             |

V. CONCLUSION

As a novel swarm intelligence-based algorithm, WOA has a simple structure and a strong ability to solve optimization problems. However, it has the defects of too many random parameters and stagnant local optimal values. Therefore, this paper first introduces non-linear adaptive weights so that the location update mechanism can adaptively transition between the exploration and exploitation stages. In addition, this paper introduces an improved golden sine operator. The search agent explores the search space efficiently with a sinusoidal route, which improves the global exploration capability of the algorithm. At the same time, the golden section coefficient enables the search agent to update the distance and direction with a fixed step size, and search in areas that only produce good results. In addition, the increase of the indication coefficient enhances the moving effect of the search agent. Search agents can quickly develop into excellent areas so as to improve the local exploitation capability of the algorithm. The simulation experiments use 28 test functions to evaluate the performance of the improved WOA (NGS-WOA). Experimental results show that the improved strategy proposed in this paper effectively improves the optimization performance of NGS-WOA. Also it is more competitive than other algorithms. At the same time, NGS-WOA can effectively solve high-dimensional optimization and engineering

optimization problems, and it has significance for further research and application of improved methods of whale optimization algorithm.

REFERENCES

[1] J. H. Holland, “Genetic algorithms,” Sci. Amer., vol. 267, no. 1, pp. 66–73, 1992.
[2] I. Rechenberg, “Evolutionstrategien,” in Simulationmethoden in der Medizin und Biologie. Berlin, Germany: Springer, 1978, pp. 83–114.
[3] R. Storn and K. D. Price, “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,” J. Global Optim., vol. 11, no. 4, pp. 341–359, 1997.
[4] D. Simon, “Biogeography-based optimization,” IEEE Trans. Evol. Comput., vol. 12, no. 6, pp. 702–713, Dec. 2008.
[5] D. Dasgupta and Z. Michalewicz, Evolutionary Algorithms in Engineering Applications. Berlin, Germany: Springer, 2013, doi: 10.1007/978-3-662-03423-1.
[6] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[7] X. S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” Int. J. Bio-Inspired Comput., vol. 2, no. 2, pp. 78–84, 2010.
[8] X. S. Yang, “A new metaheuristic bat-inspired algorithm,” in Nature inspired cooperative strategies for optimization (NICSO 2010). Berlin, Germany: Springer, 2010, pp. 65–74.
[9] X. L. Li, “A new intelligent optimization-artificial fish swarm algorithm,” M.S. thesis, Zhejiang Univ., Zhejiang, China, 2003.
[10] A. Askarzadeh and A. Reza zadeh, “A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer,” Int. J. Energy Res., vol. 37, no. 10, pp. 1196–1204, Aug. 2013.
[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Adv. Eng. Softw., vol. 69, pp. 46–61, Mar. 2014.
[12] R. Salgotra and U. Singh, “The naked mole-rat algorithm,” Neural Comput. Appl., vol. 31, no. 12, pp. 8837–8857, Dec. 2019.
[13] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC), Dec. 2009, pp. 210–214.
[14] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,” Adv. Eng. Softw., vol. 114, pp. 163–191, Dec. 2017.
[15] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimization algorithm: Theory and application,” Adv. Eng. Softw., vol. 105, pp. 30–47, Mar. 2017.
[16] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” Knowl.-Based Syst., vol. 89, pp. 228–249, Nov. 2015.
adv. eng. softw. j. zhang, j. s. wang: improved woa based on nonlinear adaptive weight and golden sine operator

[21] a. h. gandomi and a. h. alavi, “krill herd: a new bio-inspired optimization algorithm,” comman. nonlinear sci. numer. simul., vol. 17, no. 12, pp. 4831–4845, dec. 2012.

[22] p. c. pinto, t. a. runkler, and j. m. c. sousa, “wasp swarm algorithm for dynamic max-sat problems,” in proc. int. conf. adapt. natural comput. algorithms. berlin, germany: springer, 2007, pp. 350–357.

[23] x. lu and y. zhou, “a novel global convergence algorithm: bee collecting pollen algorithm,” in proc. int. conf. intell. comput. berlin, germany: springer, 2008, pp. 518–525.

[24] a. macherino, o. seref, o. seref, o. e. kundakcioglu, and p. pardalos, “monkey search: a novel metaheuristic search for global optimization,” in proc. aip comput., vol. 2007, no. 1, pp. 162–173.

[25] a. hatamloiu, “black hole: a new heuristic optimization approach for data clustering,” inf. sci., vol. 222, pp. 175–184, feb. 2013.

[26] r. a. formato, “central force optimization,” prog. electromagn. res., vol. 77, pp. 425–491, aug. 2007.

[27] e. rashidi, h. nezamabadi-pour, and s. saryazdi, “gsa: a gravitational search algorithm,” int. j. inf. sci., vol. 179, no. 13, pp. 2232–2248, jun. 2009.

[28] y. tan and y. zhu, “fireworks algorithm for optimization,” in proc. int. conf. swarm intell. berlin, germany: springer, 2010, pp. 355–364.

[29] e. atashpaz-gargari and c. lucas, “imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition,” in proc. ieee cong. evol. comput., sep. 2007, pp. 4661–4667.

[30] n. moosavian and b. kasaei roodsari, “soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks,” swarm evol. comput., vol. 17, pp. 14–24, aug. 2014.

[31] n. moosavian and b. k. roodsari, “soccer league competition algorithm, a new method for solving systems of nonlinear equations,” int. j. intell. sci., vol. 4, no. 1, pp. 7–16, 2014.

[32] n. moosavian, “soccer league competition algorithm for solving knapsack problems,” swarm evol. comput., vol. 20, pp. 14–22, feb. 2015.

[33] y. shi, “an optimization algorithm based on brainstorming process,” in emerging research on swarm intelligence and algorithm optimization. hershey, pa, usa: igi global, 2015, pp. 1–35.

[34] y. shang, x.-n. he, and h. lei, “electromagnetism-like mechanism algorithm for global optimization,” j. comput. appl., vol. 30, no. 11, pp. 2914–2918, dec. 2010.

[35] k. m. passino, “bionomicry of bacterial foraging for distributed optimization and control,” ieee control syst. mag., vol. 22, no. 3, pp. 52–67, jun. 2002.

[36] h. shayeghi and j. dadashpour, “anarchic society optimization based pid control of an automatic voltage regulator (avr) system,” elect. electron. eng., vol. 2, no. 4, pp. 199–207, aug. 2012.

[37] f. ramezani and s. lotfi, “social-based algorithm (sba),” appl. soft comput., vol. 13, no. 5, pp. 2837–2856, may 2013.

[38] f. Glover, “Tabu search—Part I,” ORSA j. comput., vol. 1, no. 3, pp. 190–206, 1989.

[39] m. pelikan, d. e. goldberg, and e. cantú-paz, “boa: the bayesian optimization algorithm,” in proc. 1st annu. conf. genetic evol. comput., vol. 1. san mateo, ca, usa: morgan kauffmann, 1999, pp. 525–532.

[40] m. pelikan and d. e. goldberg, “hierarchical boa solves Ising spin glasses and maxsat,” in proc. genetic evol. comput. conf. Berlin, Germany: springer, 2003, pp. 1271–1282.

[41] s. mirjalili, “SCA: A sine cosine algorithm for solving optimization problems,” Knowl.-Based Syst., vol. 96, pp. 120–133, mar. 2016.
[66] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” Appl. Math. Comput., vol. 188, no. 2, pp. 1567–1579, May 2007.
[67] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, “A heuristic particle swarm optimizer for optimization of pin connected structures,” Comput. Struct., vol. 85, nos. 7–8, pp. 340–349, Apr. 2007.
[68] A. Kaveh and M. Khayatadaz, “A new meta-heuristic method: Ray optimization,” Comput. Struct., vols. 112–113, pp. 283–294, Dec. 2012.

J. Zhang is currently pursuing the bachelor’s degree with the School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan, China. His main research interest is intelligent optimization algorithms.

J. S. Wang (Member, IEEE) received the B.Sc. and M.Sc. degrees in control science from the University of Science and Technology Liaoning, China, in 1999 and 2002, respectively, and the Ph.D. degree in control science from the Dalian University of Technology, China, in 2006. He is currently a Professor and a Master’s Supervisor with the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests are modeling of complex industry process, intelligent control, and computer integrated manufacturing.

* * *

J. S. Wang is currently pursuing the bachelor’s degree with the School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan, China. His main research interest is intelligent optimization algorithms.