Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 8160121, 18 pages
https://doi.org/10.1155/2023/8160121

Research Article

CMAIS-WOA: An Improved WOA with Chaotic Mapping and Adaptive Iterative Strategy

Chao-Hsien Hsieh,[1] Qing Zhang ,[2] Ya Xu,[3] and Ziyi Wang[3]

[1]*GCTB-NSU Joint Institute of Technology, Guangzhou College of Technology and Business, Foshan, Guangdong 528138, China*
[2]*Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China*
[3]*School of Cyber Science and Engineering, Qufu Normal University, Qufu, Shandong 273165, China*

Correspondence should be addressed to Qing Zhang; q_zhang163@163.com

Received 13 September 2022; Revised 10 October 2023; Accepted 18 October 2023; Published 19 December 2023

Academic Editor: Giulio E. Cantarella

This paper proposes an improved whale optimization algorithm with chaotic mapping and adaptive iteration strategy (CMAIS-WOA). This algorithm addresses the issues of the WOA algorithm that is prone to local optimal solutions with low stability. CMAIS-WOA utilizes chaotic mapping to enhance the diversity and coverage of the initial population. Also, it adaptively adjusts the weight values based on the current distribution of whale populations and the fitness of search agents. In addition, CMAIS-WOA uses an improved nonlinear convergence factor to adjust the breadth-first and depth-first search during the optimization process. The performance of the proposed CMAIS-WOA is evaluated by using 13 classical benchmark functions and IEEE CEC2014 test suite. The experimental results show that CMAIS-WOA effectively improves the stability of the optimal solution and helps the algorithm to approach the global optimal solution. The method proposed in this paper contributes to the field of optimization which solves problems more powerfully and efficiently.

1. Introduction

The optimization problem has always been a hot issue in many fields such as computer science, artificial intelligence, and engineering practice. In response to this problem, many scholars have designed various intelligent algorithms inspired by biological and physical phenomena in nature and the behavior of animal groups. These algorithms include the particle swarm optimization (PSO) algorithm [1], ant colony optimization algorithm (ACO) [2], differential evolution (DE) algorithm [3], firefly algorithm (FA) [4], bat algorithm (BA) [5], grey wolf optimization (GWO) [6], gravitational search algorithm (GSA) [7], fireworks algorithm (FWA) [8], sine cosine algorithm (SCA) [9], naked mole-rat (NMR) algorithm [10], slime mould algorithm (SMA) [11], farmland fertility algorithm (FFA) [12], Harris hawks optimization (HHO) algorithm [13], cuckoo search optimization (CSO) algorithm [14], sparrow search algorithm (SSA) [15], ant lion optimizer (ALO) algorithm [16], African vultures optimization algorithm (AVOA) [17], mountain gazelle optimizer (MGO) [18], artificial gorilla troops optimizer (GTO) [19], improved gorilla troops optimizer (IGTO) [20], improved hybrid aquila optimizer and African vultures optimization algorithm (IHAOAVOA) [21], and enhanced honey badger algorithm (EHBA) [22]. They provide powerful tools for the optimal solution of complex functions. Among them, the whale optimization algorithm (WOA) [23] is a metaheuristic algorithm proposed by Australian scholars in 2016, which is a kind of intelligent optimization algorithm. The algorithm realizes the optimization of the objective function by simulating the foraging behavior of whales in the ocean. Due to its advantages of good optimization performance and few control parameters, it has been widely applied in many fields, such as capacitor optimization site selection [24], CO_2

emission prediction [25], clinical medicine [26], image segmentation [27], and power system [28]. WOA has great potential in the fields of engineering, medicine, and economics. Then, more scholars have carried out detailed research on this algorithm.

Although the WOA algorithm has achieved results in many projects, it has still some flaws [29]. Therefore, scholars have proposed many measurement technologies to make up for the shortcomings in swarm intelligence algorithms. Salgotra et al. [30] proposed three different modified algorithms of WOA to improve its explorative ability. Zhang and Wang [31] proposed an improved WOA based on nonlinear adaptive weight and golden sine operator (NGS-WOA). This algorithm introduced nonlinear adaptive weights, which enabled the search agent to adaptively explore the search space and balance the development and exploration phases. Also, it improved the defects of low accuracy and slow convergence speed of WOA. Jiang et al. [32] proposed an improved WOA (IWOA), which combined crossover and mutation operations in DE with the whale optimization algorithm. IWOA has higher efficiency, faster convergence speed, better accuracy, and stability of solution. Fan et al. [33] proposed a new improved WOA with a joint search mechanism called JSWOA, which can maintain the diversity of the initial population for global search. Also, a new adaptive inertia weight was given to improve the convergence accuracy and speed while jumping out of the local optimum. Tang et al. [34] proposed a WOA mixed with an artificial bee colony (ACWOA). It integrated an artificial bee colony algorithm and chaotic mapping, which effectively avoided the local optimum and improved the quality of the initial solution. Li et al. [35] proposed a chaotic strategy-based opposition-based learning adaptive variable speed WOA (CQAWOA). This algorithm was used to solve the problem of insufficient convergence accuracy and convergence speed of WOA. Oliv et al. [36] proposed the chaotic WOA (CWOA) to help WOA and avoid approaching local optimal solutions.

The above works of literature have extensively studied the problems of low efficiency, low accuracy of solution, and poor local optimal ability of WOA, respectively. However, these studies did not comprehensively consider the distribution of the population and the fitness of searching agents when designing the improvement methods. In addition, some improvement algorithms are too complex to reduce their time complexity. Therefore, this paper proposes CMAIS-WOA, which is a new algorithm to solve the mentioned problems. CMAIS-WOA can improve the defect of the low stability of optimal solution when WOA runs many times. At the same time, it can solve the problem that the search agent is easy to fall into the local optimum. Table 1 shows the comparison of various characteristics between the proposed method and related algorithms.

As shown in Table 1, the superiority of CMAIS-WOA lies in the integrated consideration of factors such as population distribution and fitness, which makes the optimization process more efficient. Moreover, the algorithm is concise in its steps and easy to implement and apply. In addition, it integrates and unifies important concepts involved in literature such as [31, 36]. Also, it can achieve diversified search and find the optimal solution. In the field of navigation, this algorithm can be used for the design of bow lines of the ship. The various physical parameters of the bow are used as position components of searching agents. Also, corresponding upper and lower bounds are set based on their actual physical meanings to find the optimal design solution. Therefore, CMAIS-WOA can help designers to design bow lines with better shape and lower resistance.

The contributions of this paper are as follows.

(1) This paper proposes an improved whale optimization algorithm with chaotic mapping and adaptive iterative strategy

(2) This paper uses chaotic mapping to improve the defects of insufficient diversity and poor coverage of the population during initialization, which improves the stability of the optimal solution

(3) In this paper, WOA is improved by using an adaptive iterative strategy and an improved nonlinear convergence factor to make it approach the global optimal solution

(4) The proposed algorithm can avoid the excessive increase in the complexity of the algorithm because it is relatively simple

(5) The experimental results demonstrate the effectiveness of the proposed algorithm

The rest of this article is organized as follows. Section 2 describes the related work. Section 3 introduces the basic principles of WOA. Section 4 introduces the newly proposed algorithms CMAIS-WOA. Section 5 verifies the proposed scheme through experiments. Finally, Section 4 represents the newly proposed algorithms CMAIS-WOA.

2. Related Work

Gharehchopogh [37] studies and analyzes somequantum-inspired metaheuristic algorithms. This paper also introduces many optimization algorithms. Li et al. [38] proposed a modified whale optimization algorithm (MWOA) with multistrategy mechanism. MWOA improved the initialization, control parameters, and search strategy of WOA. This is able to solve the problems of slow convergence and local optimality of WOA. MWOA facilitates spatial search using the Lévy antiperturbation strategy. This can enhance the global search capability. Uzer and Inan [39] proposed five hybrid algorithms to improve the whale optimization algorithm. These algorithms are mainly developed by combining WOA and PSO algorithms. These algorithms explore the search space of the problem more efficiently and then avoid local optimization. Fan et al. [40] proposed a novel hybrid metaheuristic algorithm ESSAWOA. ESSAWOA combined salp swarm algorithm (SSA) and lens opposition-based learning (LOBL) strategy to enhance WOA. ESSAWOA can solve the global optimization problem efficiently.

Table 1: Comparison of algorithm characteristics.

Characteristics	Algorithm								
	NGS-WOA	IWOA	JSWOA	ACWOA	CQAWOA	CWOA	LWOA	MPMDIWOA	CMAIS-WOA
Avoid local optima	√	√	√	√	√	√	√	√	√
Integrated population distribution and fitness									√
Diversified search			√		√				√
Increase initial population diversity			√			√			√
Stability of optimal solution		√	√			√			√
Simple steps									√

3. Basic Principles of WOA

Inspired by the group hunting behavior of humpback whales in nature, WOA is proposed to better solve the function optimization problem. In terms of function optimization, this algorithm has better convergence speed and convergence accuracy than DE, PSO, GSA, and other algorithms. Also, the algorithm is easy to be implemented by few parameters to be set only.

3.1. WOA. Humpback whales are tropical marine creatures that prefer to travel in groups and cooperate to capture prey by making bubble nets. When the number of prey is relatively small, humpback whales generally hunt alone or in groups of 2 to 3. When the number of prey is relatively large, humpback whales will form a team to hunt. The team consists of about 60 whales at most.

Based on the bubble-net attaching behavior of humpback whales, WOA [23] was proposed. WOA is divided into two stages, which are the bubble-net attacking stage and the search for prey stage. Assume that the space for searching food is d-dimensional space. The number of whales in the prey group is M. Then, the position of the *j*-th whale in the *d*-dimensional space can be expressed as $\overrightarrow{X_j} = X_j^1, X_j^2, \ldots, X_j^d, j = 1, 2, \ldots, M$. Meanwhile, the current optimal position is expressed as $\overrightarrow{X^*} = (X^{1*}, X^{2*}, \ldots, X^{d*})$. There are multiple search agents. At each subsequent iteration, the search agents try to update their position until the target prey is found.

3.1.1. Bubble-Net Attacking. During this stage, the whale's predation behavior is divided into two forms. One is the shrinking encircling mechanism, and the other is the spiral updating position. When the whales take the former, the individual whales in the whale procession all initiate a siege towards the prey. Among them, the shrinkage encirclement mechanism is expressed as formula (1). Also, formula (2) represents the spiral updating position.

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

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{\mathrm{bl}} \cdot \cos(2\pi l) + \vec{X}^*(t), \tag{2}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}, \tag{4}$$

$$\vec{a} = 2 - \left(\frac{2t}{t_{\max}}\right). \tag{5}$$

The relevant descriptions of formulas (1) and (2) are described as follows. $\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| = D^1, D^2, \ldots, D^d$ represents the current d-dimensional space. $\overrightarrow{D'} = |\vec{X}^*(t) - \vec{X}(t)| = (D^{1'}, D^{2'}, \ldots, D^{d'})$ represents the distance between the whale individual and the optimal individual before the renewal of the position. $\vec{X}^*(t) = (X^{1t}, X^{2t}, \ldots, X^{\mathrm{dt}})$ represents the position coordinate vector of the prey. $\vec{X}(t)$ represents the coordinate vector of a whale individual in the global scope. The t represents the number of iterations. b is a constant that determines the shape of the spiral when updating the position. l represents a random number in $[-1, 1]$. $\vec{A}$ and $\vec{C}$ are coefficient vectors. $\vec{r}$ represents a random vector distributed in the range of $[0, 1]$. $t_{\max}$ is the maximum number of iterations, and $\vec{a}$ is a convergence factor.

In fact, in the real bubble-net attaching behavior, the shrinking encircling mechanism and the spiral updating position are carried out randomly with a probability of 0.5. The actual state is shown in the following formula:

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

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

3.1.2. Search for Prey. During this phase, individual whales in a whale team conduct random searches based on their location relative to each other. The search for prey is expressed as follows:

$$\vec{X}(t+1) = \overrightarrow{X_{\mathrm{rand}}} - \vec{A} \cdot \vec{D}, \tag{7}$$

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_{\mathrm{rand}}} - \vec{X}t\right|, \tag{8}$$

$$\vec{D} = \left(D^1, D^2, \ldots, D^d\right). \tag{9}$$

In formula (7), X_{rand} is the location of the current randomly selected prey from the whale population.

The overall representation of WOA is as follows:

$$X(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & |\vec{A}| < 1, p < 0.5, \\ \overrightarrow{D'} \cdot e^{\mathrm{bl}} \cdot \cos(2\pi l) + \vec{X}^*(t), & p \geq 0.5, \\ \overrightarrow{X_{\mathrm{rand}}} - \vec{A} \cdot \vec{D}, & |\vec{A}| \geq 1, p < 0.5. \end{cases} \tag{10}$$

3.1.3. Pseudocode of WOA. In order to present the logic of the WOA algorithm in more detail, its pseudocode is as follows (see Algorithm 1).

The specific steps of WOA are as follows:

Step 1. Set the relevant parameters of the algorithm.

Step 2. Randomly initialize each individual of the whale population.

Step 3. Calculate the fitness value $f(\overrightarrow{X_i})$ of each individual.

Step 4. Record the optimal individual and its position $X_j^*(t)$.

Step 5. Determine whether the termination condition is met. If satisfied, output the optimal solution; if not, return to Step 5

Input: Number of whales ($SearchAgents_{no}$), maximum number of iterations (t_{max}), dimension (d), lower and upper bounds of whale population location variables, and optimization function.
Output: Optimal solution of the objective function, optimal position of searching agent, and searching curve.

```
Randomly initialize the whale population
Initialize iteration counter t = 0
while (t < t_max) do
  for i = 1 to SearchAgents_no do
    Bring out-of-bounds whale individuals back into the boundary
    Calculate the fitness and location of the current optimal search agent
  end for
  for i = 1 to SearchAgents_no do
    Calculate r⃗
    Calculate A⃗ and C⃗
    Calculate p
    for j = 1 to d do
      if (p ≥ 0.5)
        Update the current individual position according to formula (6)
      end if
      else if (p < 0.5)
        if (|A⃗| < 1)
          Update the current individual position according to formula (1)
        else if (|A⃗| ≥ 1)
          Update the current individual position according to formula (7)
        end if
      end for
    end for
    t = t + 1
end while
```

ALGORITHM 1: WOA.

Step 6. Calculate $\overrightarrow{a}$ according to formula (5). Update $\overrightarrow{A}$ and $\overrightarrow{C}$ according to formulas (3) and (4).

Step 7. Generate a random number p in $[0, 1]$. If $p \geq 0.5$, update the position according to formula (2), then go to step 3; if $p < 0.5$, judge the size of $|\overrightarrow{A}|$. If $|\overrightarrow{A}| \geq 1$, update the position according to formula (7), then go to step 3; if $|\overrightarrow{A}| < 1$, update the position according to formula (1), and go to step 3.

4. Improved WOA

The research is shown in [41]. The quality of the initial population affects the algorithm's accuracy and convergence speed. Better diversity of the initial population is helpful to improve algorithm performance. However, WOA usually uses a random method to generate the initial population while solving optimization problems. It may result in uneven distribution of the initial population. Therefore, this paper uses chaotic mapping to initialize the population.

When the number of iterations increases, WOA tends to fall into the local optimal solution. In this paper, the population distribution and fitness scenarios are considered comprehensively. Then, an adaptive iteration strategy is designed. It can adjust the size of the weight value according to the current distribution of the whale population and the fitness of the current search agent location.

This paper proposes two improvements based on WOA. First, a chaotic map can improve the robustness of the algorithm. Second, an adaptive iterative strategy is combined with an improved nonlinear convergence factor to develop the ability of global search of the algorithm. Therefore, CM-WOA and AIS-WOA constitute a novel WOA (CMAIS-WOA). Among them, CM-WOA is the improved WOA with chaotic mapping. AIS-WOA is the improved WOA with an adaptive iterative strategy.

4.1. Improved WOA with Chaotic Mapping. Aiming at the defect of poor optimization stability of the search agent when WOA optimizes the objective function multiple times, CM-WOA uses a population initialization strategy based on an improved chaotic mapping. It tries to make the initial population cover the entire solution space as much as possible. This lays the foundation for the diversity of the population for the global search and local search of the algorithm. Moreover, the ergodic coverage is enhanced. Then, it can improve the robustness of the algorithm. The strategy is described in detail as follows.

In the field of optimization, chaotic mapping can be used to replace pseudorandom number generators, which generate chaotic numbers between 0 and 1. The use of chaotic sequences to initialize the population and other operations will have a good effect on the entire process of the algorithm. Also, it can often achieve better results than pseudorandom numbers. Currently, the most widely used chaotic map is the logistic map method. The populations treated with this method are diverse. Also, these populations can be extended to the entire solution space. Based on this, this paper further improves the method and uses it to initialize the population of CM-WOA. Regarding the improved logistic chaotic map method, formula (11) shows the chaotic matrix variables required to obtain the initialized population. Based on the specification requirements of the population matrix, formula (11) is iterated $\text{SearchAgents}_{\text{no}} \times d$ times to generate the chaos variable matrix required to initialize the population.

$$\begin{cases} u_j^{m+1} = \left|1 - 2 \times \left(u_j^m\right)^2\right|, \\ m = 1, 2, \ldots N, j = 1, 2, \ldots, M, \end{cases} \tag{11}$$

where u_j^{m+1} represents a chaotic variable whose value is distributed in [0, 1], N is the population dimension of the whale population after initialization, and M represents the number of search agents to initialize the population. Furthermore, the initialized population of the whale solution space can be obtained after each element in the chaotic matrix is converted, as shown in the following formula:

$$p_j^m = \min_j + \left(\max_j - \min_j\right) u_j^m. \tag{12}$$

In formula (12), $\min_j$ and $\max_j$ are the lower and upper bounds of the variables of whale population position, respectively. By this formula, the initial population of whales can be mapped out according to the chaotic matrix.

In order to reflect the internal logic of CM-WOA, the pseudocode of CM-WOA is shown as follows (see Algorithm 2):

Meanwhile, the execution steps of CM-WOA are as follows.

Step 1. Set the relevant parameters of the algorithm.

Step 2. Initialize each individual of the whale population based on formulas (11) and (12).

Step 3. Calculate the fitness value $f(\overrightarrow{X_i})$ of each individual.

Step 4. Record the optimal individual and its position $X_j^*(t)$.

Step 5. Determine whether the termination condition is met. If satisfied, output the optimal solution; if not, return to Step 5.

Step 6. Calculate $\overrightarrow{a}$ according to formula (5). Update $\overrightarrow{A}$ and $\overrightarrow{C}$ according to formulas (3) and (4).

Step 7. Generate a random number p in [0, 1]. If $p \geq 0.5$, update the position according to formula (2), then go to step 3; if $p < 0.5$, judge the size of $|\overrightarrow{A}|$. If $|\overrightarrow{A}| \geq 1$, update the position according to formula (7), then go to step 3; if $|\overrightarrow{A}| < 1$, update the position according to formula (1), then go to step 3.

4.2. Improved WOA with Adaptive Iterative Strategy. In order to improve the optimal ability of WOA, adaptively an adjusting method designs the weights based on population distribution and individual fitness of whales. So, the algorithm can adjust the weight value adaptively according to the current distribution of the whale population and the fitness of the current position of the search agent. AIS-WOA can solve the problem that the search agent is prone to fall into the local optimization and deviate from the global optimal solution. At the same time, an improved nonlinear convergence factor is designed to adjust the optimal ratio of the breadth and depth for WOA in the optimal process. The combination of the two schemes can improve the ability of individual whales to move closer to the global optimal solution. Both strategies are described as follows.

In order to improve the global search and local search capabilities of WOA, the inertia weight factor is changed to a function that changes with the number of iterations. The design of this function is shown in the following formulas:

$$\text{weight} = \left(d_1 \left(x_{j\text{worst}} - x_{\text{jbest}} \right) + \frac{d_2 y_j^{\text{up}} - y_j^{\text{down}}}{\text{t}} \right) \times \text{fitness factor}, \tag{13}$$

$$\text{fitness factor} = \begin{cases} X_{\text{rand}}(0,1), & d(j) \leq \overline{\text{low}}, \\ Y_{\text{rand}}(1,2), & d(j) \geq \overline{\text{high}}, \\ 1, & \overline{\text{low}} < d(j) < \overline{\text{high}}. \end{cases} \tag{14}$$

In formulas (13) and (14), weight is the weight of dynamic adaptive changes with population distribution and the fitness of individual locations of whales. The y_j^{up} and y_j^{down} represent the upper and lower bounds of the variable. The y_j, x_{jworst}, and x_{jbest} represent the position vector of the worst whale and optimal whale in the current state of the population distribution. Also, the t represents times of iterative foraging in the current population. In addition, d_1 and d_2 are parameters of this function. For the fitness factor, in the iteration process, the fitness $d(j)$ of all individuals in the whale population is sorted from small to large. Then, this ordered population is divided into two parts, superior end and inferior end. Furthermore, the average values $\overline{\text{low}}$ and $\overline{\text{high}}$ for the fitness level of superior and inferior ends are obtained. Finally, $d(j)$, $\overline{\text{low}}$, and $\overline{\text{high}}$ are compared to get the fitness factor of the current search agent.

Based on the above, the shrinking encircling mechanism and the spiral updating position of the improved whale bubble-net attacking behavior are described as formulas (15) and (16). In addition, formulas (17) and (18) are expressions

```
Input: Number of whales (SearchAgents_no), maximum number of iterations (t_max), dimension (d), lower and upper bounds of whale
population location variables, and optimization function.
Output: Optimal solution of the objective function, optimal position of searching agent, and searching curve.
Initialize whale population based on chaotic mapping
Initialize iteration counter t = 0
while (t < t_max) do
  for i = 1 to SearchAgents_no do
    Bring out-of-bounds whale individuals back into the boundary
    Calculate the fitness and location of the current optimal search agent
  end for
  for i = 1 to SearchAgents_no do
    Calculate r
    Calculate A and C
    Calculate p
    for j = 1 to d do
      if (p ≥ 0.5)
        Update the current individual position according to formula (6)
      end if
      else if (p < 0.5)
        if (|A| < 1)
          Update the current individual position according to formula (1)
        else if (|A| ≥ 1)
          Update the current individual position according to formula (7)
        end if
    end for
  end for
  t = t + 1
end while
```

Algorithm 2: CM-WOA.

of relevant variables. Where b is a constant that determines the shape of the log arithmetic spiral, and $D_p^{'} = (D^{1'}, D^{2'}, \ldots, D^{d'})$ represents the distance between the current whale individual and the current best search agent in the d-dimensional space. $\overrightarrow{X}_i^d(t+1) = (x_i^1, x_i^2, \ldots, x_i^d)$ represents the position vector of the i-th whale in the d-dimensional space in the $t+1$-th iteration. $\overrightarrow{X}^*(t) = (x_t^1, x_t^2, \ldots, x_t^d)$ represents the coordinate vector of the optimal solution.

$$\overrightarrow{X}_i^d(t+1) = \text{weight} \cdot \overrightarrow{X}^*(t) - \overrightarrow{A} \cdot \overrightarrow{D}, \tag{15}$$

$$\overrightarrow{X}_i^d(t+1) = \text{weight} \cdot \overrightarrow{X}^*(t) + D_p^{'} \cdot e^{bl} \cdot \cos 2\pi l, \tag{16}$$

$$\overrightarrow{D} = \left|\overrightarrow{C} \cdot \overrightarrow{X}^*(t) - \overrightarrow{X}(t)\right|, \tag{17}$$

$$D_p^{'} = \left|\overrightarrow{X}^*(t) - \overrightarrow{X}(t)\right|. \tag{18}$$

Under the action of this weight value, when the search agent for a relatively good position or individual differences in the population is small, the algorithm will produce less disturbances to the current prey. This facilitates the search for the optimal solution in the current range. Conversely, if the individual differences in the population are large or the search agent deviates far from the optimal whale individual, the algorithm will give greater disturbance to the prey. Therefore, AIS-WOA can look for prey in other locations, which improves its global optimal searchability. Moreover, whales can adaptively adjust the weight value according to the current distribution and fitness of the population. Therefore, in the process of searching for prey, whales are no longer restricted to one form but adaptively select the current prey in various forms. This makes the optimal process more diverse. Furthermore, the solution accuracy of the algorithm can be improved. The optimization ability of the algorithm can be increased. Also, the algorithm can avoid missing the global optimal solution.

The heuristic algorithm of swarm intelligence includes the breadth-first search and the depth-first search in the whole optimal process. Through the analysis of the algorithm, it is found that the coefficient vector A affects the global and local search distribution in the algorithm. The change in A is determined by the convergence factor a [42]. Furthermore, this factor is improved to alleviate the problem that the algorithm is easy to fall into the local optimization. The improved convergence factor is shown in the following formula:

$$a = 2\left(1 - \left(\frac{t}{\text{Max\_iteration}}\right)^2\right). \tag{19}$$

Formula (19) can better adjust the breadth-first search and depth-first search of the algorithm in the optimal process so that the ratio of 1 : 1 will not appear during the search. This formula increases the search ratio of the algorithm to the global optimal solution. At the same time, the convergence speed and accuracy of the algorithm are improved.

The pseudocode description of AIS-WOA is as follows (see Algorithm 3):

The detailed steps of AIS-WOA are as follows:

Step 1. Set the relevant parameters of the algorithm.

Step 2. Randomly initialize the whale population.

Step 3. Calculate the fitness value $f(\overrightarrow{X_i})$ of each individual.

Step 4. Record the optimal individual and its position $X_j^*(t)$.

Step 5. Determine whether the termination condition is met. If satisfied, output the optimal solution; if not, return to Step 5.

Step 6. Calculate a according to formula (19). Update $\overrightarrow{A}$ and $\overrightarrow{C}$ according to formulas (3) and (4).

Step 7. Calculate weight based on formulas (13) and (14).

Step 8. Generate a random number p in [0, 1]. If $p \geq 0.5$, update the position according to formula (15), then go to step 3; if $p < 0.5$, judge the size of $|\overrightarrow{A}|$. If $|\overrightarrow{A}| \geq 1$, update the position according to formula (7), then go to step 3; if $|\overrightarrow{A}| < 1$, update the position according to formula (14), then go to step 3.

4.3. Improved WOA with Chaotic Mapping and Adaptive Iterative Strategy. CMAIS-WOA is a combination of CM-WOA and AIS-WOA, which solves not only the defect of the poor stability of WOA optimization but also the problem that WOA is easy to fall into local optimization. The process of CMAIS-WOA is shown in Figure 1. In the figure, the parameter m controls the selection of the algorithm. If m equals 1, the improved WOA can execute CM-WOA; otherwise, it executes AIS-WOA.

Time complexity reflects the running efficiency of an algorithm and shows an important factor to judge the performance of an algorithm. When the processing time of whale boundary conditions is $t1$, the time to calculate the fitness value of the objective function is $f(n)$. The replacement time to compare with the current optimal fitness value is $t2$, and the calculation time of the coefficient vector is $t3$. Then, the time complexity of this stage is $T2 = O\ (N\ (n \times t1 + f(n) + t2 + t3)) = \mathrm{O}\ \ (n + f(n))$. When there are $m1$ whales in the population carrying out random walk for food, $m2$ whales carrying out contraction to surround prey and $m3$ whales carrying out spiral path to attack prey. Then, the time of position to update per dimension of whales executing the three strategies is $t4$, $t5$, and $t6$, respectively. Then, the time complexity of this stage is $T3 = O\ (N(m1(n \times t4) + m2(n \times t5) + m3(n \times t6))) = \mathrm{O}\ \ (n)$. In summary, the time complexity of WOA is $T1 + T2 + T3 = O\ (n + f(n))$. Meanwhile, CMAIS-WOA is analyzed in detail, and its time complexity is $O\ (n + f(n))$. Based on the above analysis, the time complexity of CMAIS-WOA and WOA in this paper is the same. Therefore, the execution efficiency of the algorithm is not reduced well.

5. Experimental Results and Analysis

5.1. Benchmark Functions. In this paper, 13 benchmark functions $F1$–$F13$ are used to verify the effectiveness of CMAIS-WOA. These functions are roughly classified into unimodal functions and multimodal functions. Among them, the unimodal function is $F1$–$F7$. They have only one global optimal solution, which can be used to evaluate the local utilization ability and convergence speed of the algorithm. The multimodal function is $F8$–$F13$. They have multiple local optimal solutions, which can be used to evaluate the searchability of the algorithm. These benchmark functions are shown in Table 2. They all take the minimum value as the optimal value. The solution format is 30 dimensions. In this experiment, it records the results of running the benchmark functions 100 times. In this paper, the standard deviation is used to measure the optimization stability of the algorithm, and the optimal value of the results of 100 times is used to measure the optimization ability of the algorithm. The collected data are used to verify the optimization performance of CMAIS-WOA by calculating and analysing the standard deviation and the maximum value. Then, the proposed algorithm in this paper has a high performance.

The hardware environment of the computer used in this experiment is Intel(R) Core (TM) i5-8250U CPU@ 1.60 GHz 1.80 GHz. The software environment is MATLAB R2017b. In order to fully utilize the optimization performance of CMAIS-WOA, the parameters are set as follows. The constants b in formula (6) and formula (15) are set to 1. The parameters $d1$ and $d2$ in formula (12) are set to 0.005. This setting can make the experimental results better. Also, the source code to simulate is available for downloading in the GitHub link as follows: https://github.com/znlazy/WOA.

5.2. Verification of Algorithm Stability. Because it is necessary to verify the stability of the optimization results of CMAIS-WOA, this section uses the standard deviation as the performance indicator. In [31], various optimization algorithms are calculated. This paper compares PSO algorithms [1], FA [4], FWA [8], SCA [9], WOA [23], and DE [43] with CMAIS-WOA. To ensure the best results, the configuration parameters of each algorithm are set to the optimal values proposed by the author in the original article. The parameters of these algorithms are shown in Table 3. For a fair comparison, the maximum number of iterations of the seven algorithms is set to 1000. The size of the population is set to 30. The experimental results are shown in Table 4.

As shown in the table, the proposed CMAIS-WOA has the best performance on $F1$, $F2$, $F6$, and $F7$ on unimodal functions. WOA performs best on functions $F3$, $F4$, and $F5$. PSO performs well on functions $F1$, $F2$, and $F6$. FA performs well on functions $F1$ and $F6$. FWA performs well in $F1$ and $F2$. SCA does well in $F2$. DE performs well in $F13$. On multimodal functions, CMAIS-WOA performs best on

Input: Number of whales ($\text{SearchAgents}_{\text{no}}$), maximum number of iterations ($t_{\max}$), dimension (d), lower and upper bounds of whale population location variables, and optimization function.
Output: Optimal solution of the objective function, optimal position of searching agent, and searching curve.

```
Randomly initialize the whale population
Initialize iteration counter t = 0
while (t < t_max) do
  for i = 1 to SearchAgents_no do
    Bring out-of-bounds whale individuals back into the boundary
    Calculate the fitness and location of the current optimal search agent
  end for
  Calculate the nonlinear convergence factor a
  for i = 1 to SearchAgents_no do
    Calculate r→
    Calculate A→ and C→
    Calculate p
  Calculate the improved inertia weight factor
    for j = 1 to d do
      if (p ≥ 0.5)
        Update the current individual position according to formula (16)
      end if
      else if (p < 0.5)
        if (|A→| < 1)
          Update the current individual position according to formula (15)
        else if (|A→| ≥ 1)
          Update the current individual position according to formula (7)
        end if
      end for
    end for
    t = t + 1
end while
```

Algorithm 3: AIS-WOA.

functions $F9$, $F10$, $F11$, $F12$, and $F13$. WOA performs best on function $F8$. FA does well in $F12$ and $F13$. FWA is the best performer in $F11$. Therefore, CMAIS-WOA performs well on 69.2% of the functions. WOA and FA perform well on 30.8% of the functions. PSO and FWA perform well on 23.08% of the functions. SCA and DE perform well on the 7.7% function. Therefore, the improved strategy proposed in this paper can effectively improve the convergence speed of WOA and the stability of the optimal solution.

5.3. Verification of Algorithm Optimization Ability. The optimization performance of CMAIS-WOA is evaluated by running CMAIS-WOA and other optimization algorithms on 13 benchmark functions 100 times, respectively. During the whole iterative process, the convergence factor a and the coefficient variable A control the breadth-first search and depth-first search of whale foraging behavior. A is the controller of the search pattern for the entire optimization process. By improving it, CMAIS-WOA changes the 1 : 1 optimization ratio of breadth-first search and depth-first search. In Figure 2, CMAIS-WOA performs an extensive global search while iterating. It preserves the diversity of the population. In this figure, A1 represents before improvement, and A2 represents after improvement.

In this experiment, the minimum value is used as the performance indicator. It is used to express the optimization ability of the algorithm on the corresponding function. The comparison results are shown in Table 5.

In Table 5, CMAIS-WOA has the best performance on functions $F1$, $F2$, $F3$, $F4$, $F7$, $F8$, $F9$, and $F10$. Also, it performs well on function $F6$. WOA performs well on functions $F1$, $F2$, $F3$, $F4$, and $F10$. PSO and FA perform well on functions $F11$, $F12$, and $F13$. FWA performs best on functions $F9$ and $F11$. DE performs well on function $F13$. Therefore, CMAIS-WOA finds the theoretical optimum on the 69.2% function. WOA found the theoretical optimum on the 38.5% function. PSO and FA find the theoretical optimum on the 23.08% function. FWA finds the theoretical optimum on the 15.4% function. DE finds the theoretical optimum on the 7.7% function. Moreover, CMAIS-WOA is 30.7% higher than WOA in the global optimal solution search efficiency. Therefore, the improved WOA can improve the optimization ability of the algorithm. It is worth mentioning that CMAIS-WOA performs well on the same number of benchmark functions in this experiment and the verification experiment of the algorithm stability, all of which are 9. However, the types of these 9 benchmark functions are different.

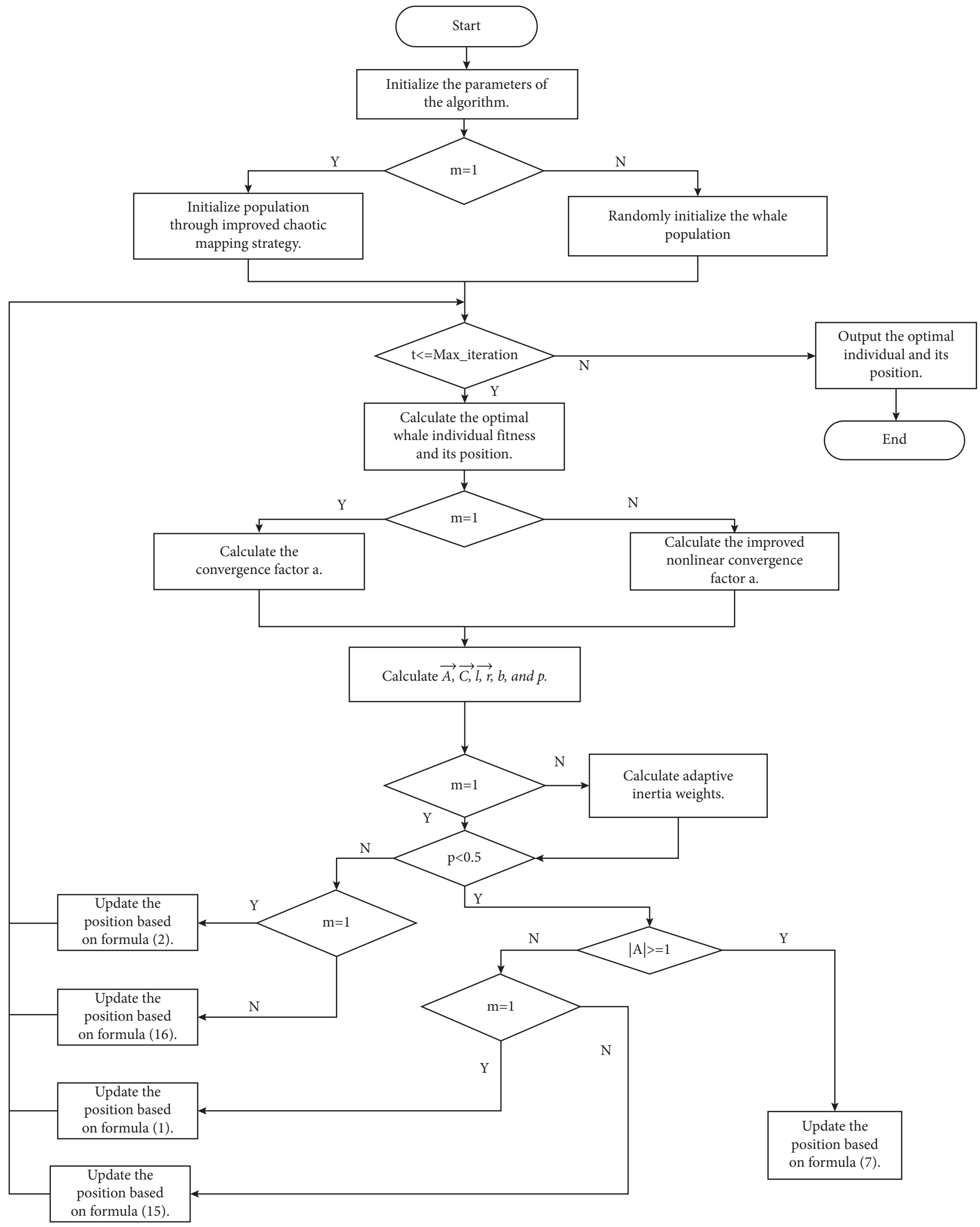


Figure 1: The process of CMAIS-WOA.

Figure 3 shows the benchmark function iteration process with good performance of CMAIS-WOA. It records the iterative trajectory of the search agent to find the global optimal solution. The metaheuristic algorithm makes an improved design for the inertia weight factor value and nonlinearly adjusts the convergence factor. Therefore, it has a good breadth-first search and depth-first search layout structure throughout the optimization process. First, the breadth-first search is significant, which guarantees the algorithm priority in global search. Second, the convergence speed is fast, and the global optimal solution is efficiently approached.

Figures 3(a), 3(c), 3(e), 3(g), 3(i), 3(k), 3(m), 3(o), and 3(q) are 3D contour maps of the functions $F1$, $F2$, $F3$, $F4$, $F6$, $F7$, $F8$, $F9$, and $F10$ based on CWAIS-WOA, respectively. From this figure, the optimal value for distributions of functions $F1$, $F2$, $F3$, $F4$, $F6$, and $F10$ is relatively concentrative. Therefore, they are not easily misled by partial solutions. The optimal values of functions $F7$, $F8$, and $F9$ are scattered and easily misled by partial solutions. Figures 3(b), 3(d), 3(f), 3(h), 3(j), 3(l), 3(n), 3(p), and 3(r) are the graphs of the iterative process of the above function, respectively. They show the process of the function converging to the global optimal solution. For function $F1$, CMAIS-WOA

Table 2: Objective functions.

Function	V_{no}	Range	$F(x)_{min}$
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[−10, 10]	0
$F_3(x)=\sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100, 100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert, 1< =i< =n\}$	30	[−100, 100]	0
$F_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i)^2+(x_i-1)^2]$	30	[−30, 30]	0
$F_6(x)\sum_{i=1}^{n}([x_i+0.5])^2$	30	[−100, 100]	0
$F_7(x)=\sum_{i=1}^{n} ix_i^4+\text{random}[0,1)$	30	[−1.28, 1.28]	0
$F_8(x)=\sum_{i=1}^{n} -x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[−500, 500]	−418.9829 × 30
$F_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[−5.12, 5.12]	0
$F_{10}(x)=-20\exp(-0.2\sqrt{1\sum_{i=1}^{n} x_i^2/n})-\exp(1/n\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[−32, 32]	0
$F_{11}(x)=1/400\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1$	30	[−600, 600]	0
$F_{12}(x)=\pi/n\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2\pi y_{i+1}]+(y_n-1)^2\}+\sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i=1+x_i+1/4u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m, x_i>a \\ 0-a<x_i<a \\ k(-x_i-a)^m x_i<-a \end{cases}$	30	[−50, 50]	0
$F_{13}(x)=0.1\{\sin^2 3\pi x_1+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n} u(x_i,5,100,4)$	30	[−50, 50]	0

Table 3: Parameter settings for algorithms.

Algorithms	Parameter settings
CWAIS-WOA	$B=1$, $d1$ and $d2=0.005$
WOA	$r=(0,1)$, $a=(0,1)$
PSO	$w=0.7298$, c_1 and $c_2=1.4962$
FA	gamma = 1, alpha = 0.2, beta 0 = 2, $m=2$
FWA	$n=5$, $\widehat{A}=40$, $M_e=150$, $N_{min}=6$, $N_{max}=120$
SCA	$a=2$
DE	$F=0.5$, Cr = 0.5

Table 4: Comparison results of algorithm stability.

Function	CWAIS-WOA	WOA	PSO	FA	FWA	SCA	DE
$F1$	0	0.304185	$4.41E-05$	$3.69E-09$	$2.32E-35$	$4.86E+03$	$8.69E+03$
$F2$	0	0.251213	$8.07E-59$	$1.07E-04$	$7.11E-19$	$5.13E-09$	$1.08E+19$
$F3$	2.232	0	$1.12E+06$	$4.02E+04$	$2.96E+02$	$6.66E+08$	$7.18E+04$
$F4$	2.2533	$2.77546E-11$	$2.85E+04$	$1.45E-03$	$1.05E-03$	$1.94E+06$	$1.90E+00$
$F5$	0.532503	$2.70754E-18$	$8.01E+06$	$2.33E+05$	$2.58E+03$	$5.52E+06$	$6.23E+07$
$F6$	$1.37867E-16$	0.282224	$4.68E-10$	$2.15E-09$	$3.51E+06$	$3.74E+05$	$9.00E+03$
$F7$	$3.997E-24$	0.168683	$3.81E+07$	$1.48E+03$	$4.59E+06$	$7.92E+03$	$4.16E+01$
$F8$	13.341	$2.32227E-16$	$3.69E+08$	$1.32E+08$	$4.47E+08$	$9.90E+07$	$3.12E+01$
$F9$	0	0.217932	$6.66E+06$	$2.04E+06$	$4.58E+06$	$6.84E+06$	$1.02E-01$
$F10$	$2.13E-15$	0.236827	$2.78E+00$	$7.17E-04$	$4.59E+06$	$2.85E+06$	$5.31E-02$
$F11$	0	0.117082	$1.64E+03$	$1.33E+03$	0	$5.40E+04$	$2.12E+08$
$F12$	$3.2922E-21$	0.251704	$1.01E+04$	$8.58E-12$	$7.08E+04$	$1.19E+05$	$2.26E+08$
$F13$	$1.93716E-21$	0.224765	$1.59E+03$	$1.25E-10$	$5.31E+04$	$9.42E+05$	$1.21E-20$

searches in a relatively large range before the 330th iteration. After the 330th iteration, it began to conduct a more precise search. The speed of updating the optimal search agent is very fast. Therefore, the cut-off point for the number of iterations in $F1$ is 330. Moreover, the cut-off points for the number of iterations in $F2$, $F3$, $F4$, $F6$, $F7$, $F8$, $F9$, and $F10$ are 470, 350, 340, 320, 180, 20, 17, and 370, respectively. Additionally, Table 6 shows the final optimal search agent positions for these benchmark functions.

Furthermore, the position of each function falling to the lowest point in the graph is analyzed and summarized. The curve of function $F2$ drops last. It is around the 470th

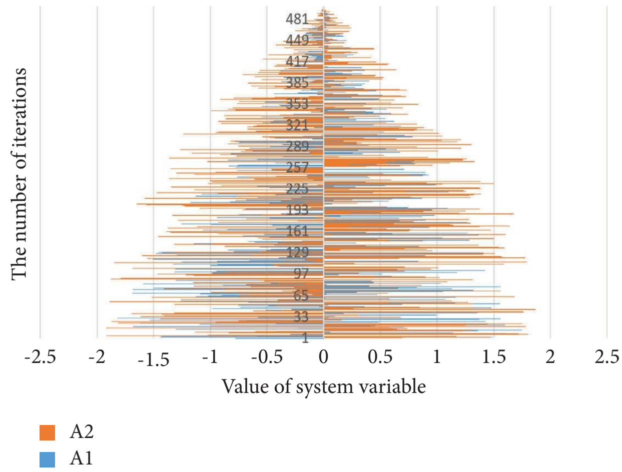


Figure 2: Comparison of coefficient variables before and after the improvement.

Table 5: Comparison results of algorithm optimization ability.

Function	CWAIS-WOA	WOA	PSO	FA	FWA	SCA	DE
$F1$	0	$4.07E-80$	$5.26E-11$	$1.08E-11$	$8.74E-53$	$1.68E-03$	$1.21E+05$
$F2$	$3.77E-191$	$1.39E-83$	$2.60E-66$	$3.26E-09$	$1.80E-33$	$5.94E-06$	$2.56E+18$
$F3$	0	$9.37E-84$	$6.01E+02$	$4.56E-11$	$9.54E-31$	$6.98E+04$	$2.03E+05$
$F4$	$3.0187e-321$	$6.76E-84$	$4.18E+01$	$2.16E-05$	$4.54E-17$	$2.26E+03$	$7.12E+01$
$F5$	0.35574	$2.60+01$	$3.54E+03$	$4.76E+03$	$5.80E+03$	$5.78E+03$	$2.10E+07$
$F6$	$4.70E-10$	0.17205	$1.09E-10$	$1.48E-11$	$1.31E+03$	$7.08E+02$	$1.12E+04$
$F7$	$5.76E-06$	$1.31E-04$	$1.92E+04$	$4.52E+00$	$3.02E-02$	$1.96E+00$	$2.13E+01$
$F8$	−3819.3924	−8706.7555	$-1.47E+06$	$-2.09E+06$	$-6.69E+05$	$-9.85E+05$	$3.40E+01$
$F9$	0	0	$3.78E+03$	$5.58E+03$	0	$2.74E-02$	$2.07E+00$
$F10$	$8.88E-16$	$8.88E-16$	$1.49E-04$	$1.23E-05$	$1.78E-13$	$3.82E-02$	$2.04E-05$
$F11$	$4.58E-03$	0	$5.18E-12$	$2.10E-11$	0	$8.80E-02$	$1.12E+03$
$F12$	$5.63E-05$	$3.11E-04$	$5.98E-12$	$3.22E-14$	$1.79E+02$	$6.72E+01$	$2.02E+03$
$F13$	$2.34E-07$	$3.11E-03$	$1.76E-11$	$4.66E-13$	$1.87E+02$	$3.90+01$	$0.35E-10$

iteration. Also, the curve of function $F9$ drops earliest. It is around the 17th iteration. The number of iterations for the former is about 27 times that of the latter. This shows that the amount of computation required by function 2 is relatively large, but, at the same time, the search range is large. On the contrary, the function $F9$ requires a relatively small amount of computation. However, it also shows that the search scope of this function is relatively small.

5.4. Comparison of Algorithm Performance on CEC2014. In order to further verify the performance and effectiveness of CMAIS-WOA algorithm, the CEC2014 benchmark set proposed by [44] is the optimal result. In the comparative experiment, all algorithms adopt the same experimental parameters for the fairness of comparison. The population size is 30, and the number of iterations is 30000. For each test function, each algorithm is run independently for 50 times. Its average accuracy and standard deviation are recorded. The results are shown in Table 7. CMAIS-WOA performs well in functions $F5$, $F6$, $F9$–$F16$, $F18$, $F19$, $F22$, $F23$, $F24$, $F26$, and $F28$–$F30$. He performed well in 19 functions and ranked first. Therefore, CMAIS-WOA has the best performance to compare with all other algorithms.

In view of the shortcomings of the WOA algorithm, this paper will improve it from two aspects. First, the chaotic map makes the initial population cover the entire solution space as much as possible. Also, the stability of the optimal solution can be improved. Second, the adaptive iterative strategy can adaptively adjust the weight value according to the current distribution of the whale population and the fitness of the current search agent location. At the same time, the improved nonlinear convergence factor can adjust the proportion of breadth search and depth search in the optimal process. The combination of the two schemes further improves the optimization performance of the algorithm. The experimental results show that CMAIS-WOA has better optimization performance, high accuracy, and fast convergence compared with other algorithms.

CWAIS-WOA still has two certain limitations, even though it performs well in optimization issues. First, CWAIS-WOA may be affected by the curse of dimensionality, thus reducing its effectiveness in the case of high-dimensional optimization. Second, the features of the optimization issue could restrict the algorithm's application. To address these limitations, further improvements can be made to CWAIS-WOA. For example, dimensionality

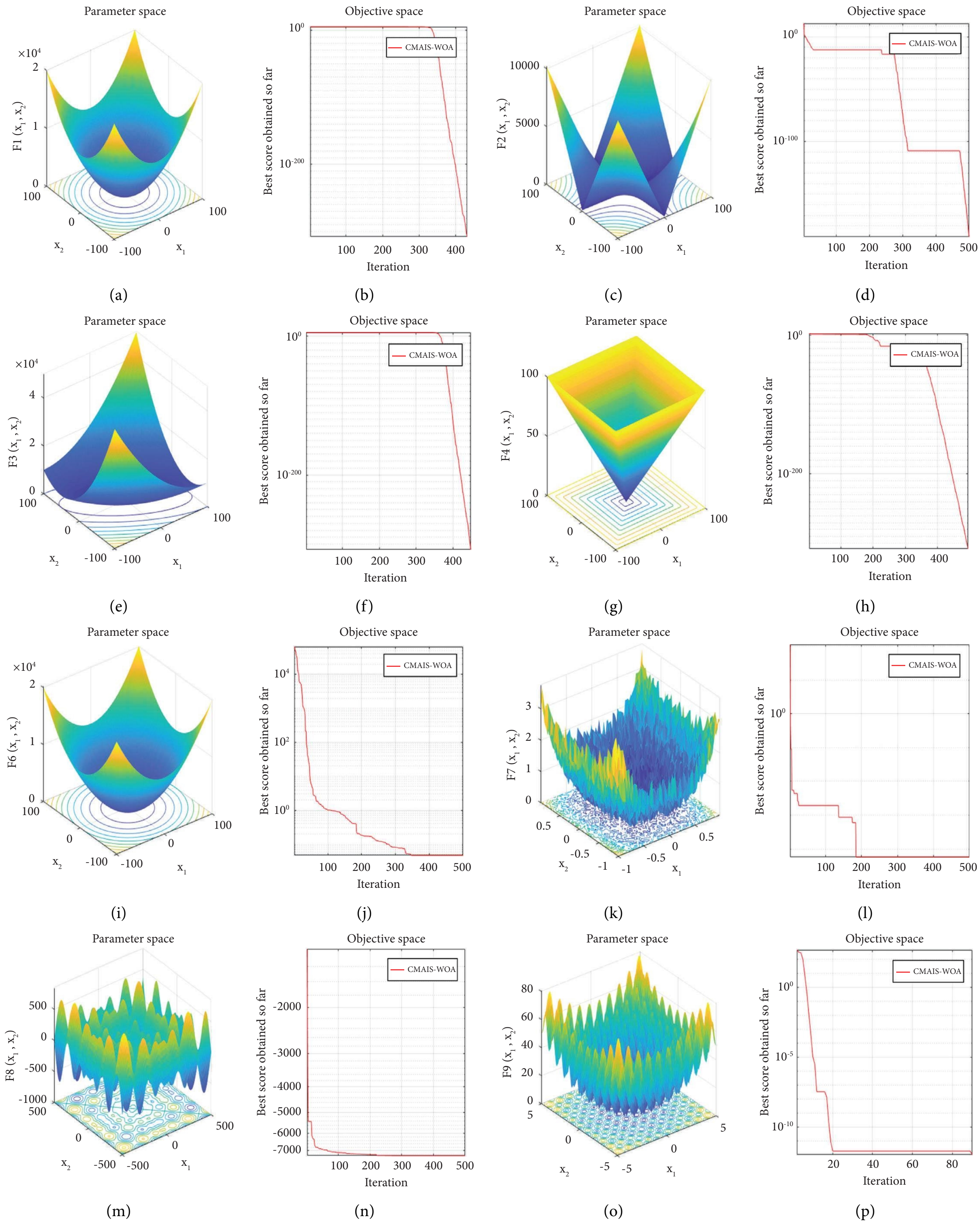
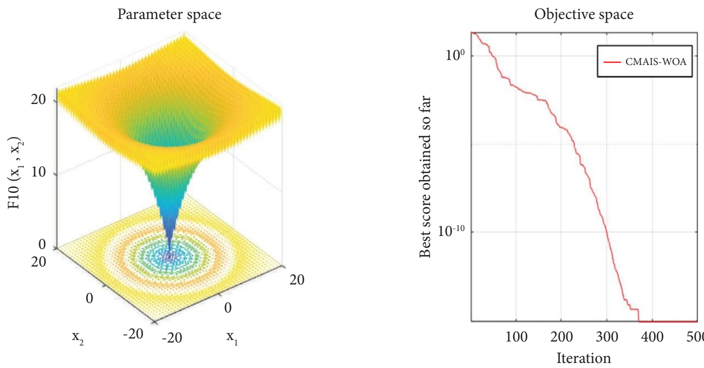


Figure 3: Continued.



(q) (r)

FIGURE 3: The optimization curve of objective functions. (a, b) F1. (c, d) F2. (e, f) F3. (g, h) F4. (i, j) F6. (k, l) F7. (m, n) F8. (o, p) F9. (q, r) F10.

TABLE 6: The location of the optimal search agent.

Functions	The location of the optimal search agent
*F*1	−4.7275*E* − 164, −2.5323*E* − 164, −7.7747*E* − 165, −3.3654*E* − 166, −2.9388*E* − 165, −6.0893*E* − 166, −3.006*E* − 165, −6.667*E* − 165, −5.4872*E* − 164, −6.3437*E* − 165, −1.7266*E* − 164, −5.5087*E* − 165, −2.0276*E* − 165, −1.0011*E* − 164, −3.8337*E* − 165, −8.1942*E* − 165, −4.6627*E* − 165, −8.3872*E* − 164, −4.344*E* − 165, −3.3592*E* − 164, −4.9437*E* − 165, −6.8899*E* − 165, −1.5656*E* − 163, −4.6584*E* − 165, −1.2529*E* − 164, −4.1324*E* − 165, −1.3557*E* − 164, −2.4771*E* − 164, −3.7056*E* − 165, −5.0894*E* − 166
*F*2	−1.1899*E* − 194, −4.0516*E* − 194, −1.5241*E* − 194, −3.9174*E* − 195, −5.0011*E* − 195, −1.9724*E* − 194, −1.3129*E* − 192, −3.7393*E* − 195, −2.4364*E* − 194, −4.0643*E* − 194, −5.15*E* − 194, −3.124*E* − 195, −4.3301*E* − 194, −5.1193*E* − 194, −1.9933*E* − 194, −4.5056*E* − 195, −2.0946*E* − 194, −1.6244*E* − 191, −5.253*E* − 193, −5.1014*E* − 194, −8.6326*E* − 195, −4.8899*E* − 195, −2.7365*E* − 195, −2.9464*E* − 194, −5.4173*E* − 192, −9.4323*E* − 195, −1.7318*E* − 194, −1.5789*E* − 194, −1.3729*E* − 191, −7.3625*E* − 196
*F*3	8.3778*e* − 166, 1.6103*e* − 165, 1.828*e* − 165, 9.0349*e* − 166, 9.7009*e* − 165, 1.3422*e* − 164, 1.5092*e* − 166, 1.0885*e* − 165, 4.7291*e* − 166, 1.102*e* − 165, 1.9386*e* − 167, 6.4553*e* − 165, 4.8061*e* − 165, 6.7909*e* − 166, 6.4788*e* − 165, 3.8793*e* − 165, 1.8414*e* − 166, 1.6037*e* − 165, 5.8625*e* − 165, 5.076*e* − 165, 6.3804*e* − 165, 4.4564*e* − 166, 3.551*e* − 165, 2.0262*e* − 165, 1.0951*e* − 165, 2.806*e* − 165, 3.228*e* − 165, 4.738*e* − 166, 8.4409*e* − 166, 1.7447*e* − 165
*F*4	7.9051*e* − 323, 1.4822*e* − 323, 9.3872*e* − 323, 0, 9.3872*e* − 323, 4.9407*e* − 324, 5.5335*e* − 322, 3.0187*e* − 321, 2.4703*e* − 323, 5.9288*e* − 323, 0, 0, 8.3991*e* − 323, 7.9051*e* − 323, 1.0375*e* − 322, 1.6798*e* − 322, 2.9644*e* − 323, 1.334*e* − 322, 4.0019*e* − 322, 1.8774*e* − 322, 1.7292*e* − 322, 1.334*e* − 322, 9.8813*e* − 324, 1.4328*e* − 322, 2.5494*e* − 321, 1.0375*e* − 322, 4.4466*e* − 323, 7.9051*e* − 323, 5.4347*e* − 323, 3.162*e* − 322
*F*6	−0.451843212, −0.488132823, −0.502485368, −0.506691793, −0.516383667, −0.522936619, −0.528443899, −0.500528215, −0.476424727, −0.546169154, −0.459999792, −0.417339417, −0.451636971, −0.459796279, −0.492525458, −0.488044512, −0.585552524, −0.491116219, −0.512569987, −0.549531897, −0.585455076, −0.440897739, −0.508565845, −0.431517191, −0.474593199, −0.478770159, −0.529879077, −0.505666429, −0.497379624, −0.505103548
*F*7	0.00079039, 0.0024982, 0.00011346, 6.3648*E* − 05, 4.9686*E* − 05, 0.0090322, 0.0017595, 9.9808*E* − 06, 0.0002355, 0.00014075, 0.0003258, 0.0015206, 0.0023369, 4.9935*E* − 06, 0.00097185, 0.00029976, 0.00032523, 2.6181*E* − 05, 0.0066135, 0.00018268, 0.0049133, 0.00026729, 6.1492*E* − 05, 0.004145, 0.010693, 0.0014858, 9.078*E* − 05, 0.00015736, 7.5053*E* − 05, 0.00050359
*F*8	427.1183, −500, −500, −500, 427.1183, 427.1183, −500, −500, 427.1183, −500, −500, −500, −500, −500, −500, 427.1183, −500, −500, −500, −500, −500, −500, 427.1183, −500, −500, −500, −500, −500, −500, −500

Table 6: Continued.

Functions	The location of the optimal search agent
$F9$	−0.000000000021882, −0.00000000023497, −0.000000000019307, −0.000000000024775, −0.00000000019519, −0.000000000035178, −0.000000000048885, −0.000000000038956, −0.000000000059401, −0.00000000014523, −0.000000000052906, −0.000000000020087, −0.0000000022882, −0.000000000073019, −0.000000000077924, −0.00000000103, −0.00000000053772, −0.000000000021679, −0.000000000095406, −0.000000000072401, −0.00000000001395, −0.000000000019946, −0.0000000000016862, −0.000000000053492, −0.000000000071809, −0.00000000011175, −0.000000000008427, −0.000000000074466, −0.000000000030654, −0.000000000060641
$F10$	$3.85E-18$, $4.17E-17$, $1.06E-16$, $9.56E-18$, $1.26E-16$, $4.33E-17$, $1.27E-16$, $1.43E-18$, $1.29E-17$, $7.96E-19$, $8.40E-18$, $1.91E-17$, $2.48E-17$, $1.47E-16$, $2.65E-19$, $4.41E-17$, $4.70E-17$, $4.78E-17$, $1.05E-16$, $1.11E-16$, $1.88E-17$, $6.00E-19$, $2.02E-17$, $2.13E-18$, $5.32E-18$, $1.05E-16$, $5.90E-18$, $1.15E-17$, $2.92E-18$, $7.40E-17$

Table 7: Statistical results of the proposed algorithm in comparison to other algorithms for CEC 2014.

Function		CWAIS-WOA	WOA	PSO	FA	FWA	SCA	DE
$F1$	Ave	$2.13E+04$	$6.30E+06$	$2.40E-14$	$3.71E+07$	$1.58E+06$	$1.15E+07$	$2.12E+07$
	Std	$2.12E+04$	$1.40E+06$	$1.43E-14$	$7.12E+06$	$1.17E+06$	$7.37E+07$	$2.47E+07$
$F2$	Ave	$2.06E+02$	$2.55E+04$	$5.59E-14$	$0.00E+00$	$0.00E+00$	$9.15E+03$	$2.56E+09$
	Std	$3.60E+02$	$9.69E+03$	$2.21E-14$	$0.00E+00$	$0.00E+00$	$6.49E+04$	$1.79E+09$
$F3$	Ave	$4.90E+01$	$6.33E+03$	$1.57E-13$	$0.00E+00$	$3.83E-11$	$5.47E+02$	$1.97E+04$
	Std	$5.92E+02$	$3.45E+03$	$4.71E-14$	$1.20E-14$	$1.34E-10$	$1.61E+03$	$6.13E+03$
$F4$	Ave	$1.57E-01$	$2.02E+02$	$1.87E-13$	$4.11E+01$	$4.54E+00$	$5.86E+02$	$2.36E+02$
	Std	$2.83E+00$	$4.13E+01$	$5.21E-14$	$4.11E+01$	$1.87E+01$	$4.19E+01$	$6.45E+01$
$F5$	Ave	$4.92E-01$	$2.74E+00$	$2.10E+01$	$2.37E+01$	$2.35E+01$	$2.72E+01$	$2.14E+01$
	Std	$1.80E+00$	$3.91E-01$	$1.37E-02$	$6.80E-02$	$4.71E-02$	$1.43E-01$	$1.54E-01$
$F6$	Ave	$1.65E-01$	$1.49E+01$	$4.50E+01$	$2.18E+01$	$3.88E+00$	$2.68E+01$	$1.74E+01$
	Std	$1.40E+00$	$2.30E+00$	$1.16E+01$	$1.61E+00$	$2.91E+00$	$4.79E+00$	$2.09E+00$
$F7$	Ave	$1.61E-07$	$7.21E-02$	$2.76E-03$	$1.03E-09$	$1.23E-04$	$1.52E+00$	$1.39E+01$
	Std	$1.17E-06$	$4.14E-02$	$4.85E-03$	$7.64E-09$	$1.58E-03$	$6.07E+00$	$1.28E+01$
$F8$	Ave	$1.55E+00$	$4.73E-01$	$4.29E+02$	$1.06E+01$	$9.79E-01$	$6.13E+01$	$1.57E+02$
	Std	$8.13E+00$	$6.71E-01$	$9.05E+01$	$1.03E+01$	$1.68E+00$	$2.67E+01$	$2.74E+01$
$F9$	Ave	$5.74E+00$	$9.19E+01$	$6.37E+02$	$1.48E+02$	$3.19E+01$	$1.13E+02$	$1.39E+02$
	Std	$1.09E+01$	$1.53E+01$	$1.51E+02$	$1.19E+01$	$6.50E+00$	$4.60E+01$	$2.56E+01$
$F10$	Ave	$1.45E+01$	$6.28E+03$	$5.11E+03$	$1.02E+02$	$4.93E+01$	$1.78E+03$	$2.62E+03$
	Std	$1.04E+01$	$4.18E+02$	$8.12E+02$	$6.48E+01$	$6.09E+01$	$7.21E+02$	$6.83E+02$
$F11$	Ave	$5.38E+01$	$5.71E+03$	$4.14E+03$	$4.49E+03$	$1.43E+03$	$3.94E+03$	$3.10E+03$
	Std	$3.84E+02$	$4.17E+02$	$6.61E+02$	$1.91E+02$	$4.14E+02$	$7.32E+02$	$6.36E+02$
$F12$	Ave	$3.50E-03$	$2.11E-02$	$1.82E-01$	$1.34E+00$	$4.10E-01$	$3.71E-01$	$6.43E-01$
	Std	$2.92E-02$	$1.25E-18$	$1.61E-01$	$1.63E-01$	$1.18E-01$	$1.75E-01$	$3.76E-01$
$F13$	Ave	$4.70E-03$	$6.48E-01$	$1.35E-01$	$3.44E-01$	$1.16E-01$	$5.12E-01$	$5.52E-01$
	Std	$3.25E-02$	$7.18E-02$	$4.87E-02$	$3.80E-02$	$3.62E-02$	$1.48E-01$	$8.97E-02$
$F14$	Ave	$3.10E-03$	$3.23E-01$	$3.30E-01$	$3.15E-01$	$2.75E-01$	$4.43E-01$	$2.38E+00$
	Std	$1.93E-02$	$1.15E-01$	$7.56E-02$	$7.75E-02$	$1.13E-01$	$2.45E-01$	$3.32E+00$
$F15$	Ave	$2.52E-01$	$1.28E+01$	$3.82E+00$	$1.54E+01$	$3.39E+00$	$3.79E+01$	$8.73E+01$
	Std	$1.51E+00$	$5.24E+00$	$1.26E+00$	$9.20E-01$	$9.89E-01$	$9.20E+01$	$1.05E+02$
$F16$	Ave	$2.40E-01$	$1.06E+01$	$3.17E-01$	$2.17E-01$	$5.12E-01$	$5.22E-01$	$6.93E-01$
	Std	$1.24E+00$	$6.23E-01$	$1.25E+03$	$1.44E+06$	$1.32E+05$	$2.50E+04$	$9.46E+05$
$F17$	Ave	$2.31E+03$	$1.23E+06$	$3.38E+02$	$6.44E+05$	$1.14E+05$	$2.14E+04$	$7.32E+05$
	Std	$2.38E+04$	$5.43E+05$	$1.45E+02$	$2.55E+03$	$9.53E+02$	$3.34E+04$	$1.48E+05$
$F18$	Ave	$4.54E+00$	$8.12E+02$	$4.50E+01$	$2.15E+03$	$1.66E+03$	$1.43E+05$	$9.10E+05$
	Std	$3.11E+01$	$1.30E+03$	$9.18E+00$	$8.23E+00$	$4.71E+00$	$1.31E+01$	$2.48E+01$

Table 7: Continued.

Function		CWAIS-WOA	WOA	PSO	FA	FWA	SCA	DE
F19	Ave	1.98E − 01	7.21E + 03	1.45E + 00	7.37E − 01	8.43E − 01	1.58E + 01	1.53E + 01
	Std	1.17E + 00	4.27E + 03	2.23E + 02	2.45E + 02	2.15E + 01	3.62E + 03	4.64E + 03
F20	Ave	7.77E + 01	1.42E + 04	1.51E + 02	1.22E + 02	3.06E + 01	7.17E + 03	3.72E + 03
	Std	5.29E + 02	4.41E + 03	1.40E + 03	2.51E + 05	1.82E + 04	5.28E + 04	2.65E + 05
F21	Ave	2.58E + 02	1.12E + 06	3.68E + 02	1.71E + 05	2.55E + 04	1.44E + 05	2.49E + 05
	Std	2.16E + 03	7.56E + 05	3.10E + 02	1.82E + 02	5.98E + 01	6.51E + 02	3.59E + 02
F22	Ave	1.21E + 01	1.58E + 02	1.77E + 02	6.13E + 01	4.82E + 01	2.61E + 02	1.28E + 02
	Std	9.26E + 01	2.17E + 02	3.13E + 02	3.25E + 02	3.11E + 02	315E + 02	3.49E + 02
F23	Ave	3.91E + 00	3.73E + 02	1.75E − 13	1.46E − 12	1.33E − 12	2.20E − 01	4.53E + 00
	Std	2.85E + 01	2.14E + 01	2.92E + 02	2.45E + 02	2.24E + 02	1.46E + 02	2.20E + 02
F24	Ave	3.91E + 00	3.41E + 04	3.84E + 02	2.65E + 00	5.46E + 00	4.69E + 00	1.56E − 03
	Std	2.82E + 01	2.75E + 04	2.01E + 02	2.12E + 02	2.08E + 02	1.08E + 02	2.51E + 02
F25	Ave	3.91E + 00	6.13E + 02	2.94E + 00	1.61E + 00	9.01E − 01	3.51E + 00	2.42E + 00
	Std	2.83E + 01	6.91E + 01	1.03E + 02	1.20E + 02	1.05E + 02	1.01E + 02	1.05E + 02
F26	Ave	1.97E + 00	3.24E + 01	1.36E + 01	4.45E − 02	4.22E − 02	1.17E + 00	1.52E − 01
	Std	1.41E + 01	5.52E + 01	4.14E + 02	5.76E + 02	3.31E + 02	7.82E + 02	8.17E + 02
F27	Ave	3.97E + 00	3.14E + 02	1.78E + 02	1.18E + 02	4.40E + 01	2.33E + 02	8.12E + 01
	Std	2.83E + 01	1.80E + 02	3.91E + 03	8.49E + 02	8.55E + 02	1.49E + 03	8.68E + 02
F28	Ave	3.18E + 00	2.22E + 03	3.07E + 03	2.75E + 01	2.68E + 01	2.38E + 02	2.06E + 02
	Std	2.60E + 01	4.44E + 02	8.05E + 02	1.48E + 05	9.17E + 05	4.40E + 06	2.19E + 06
F29	Ave	3.22E + 00	3.79E + 07	9.42E + 01	1.58E + 06	2.51E + 06	4.53E + 06	2.25E + 06
	Std	2.50E + 01	6.11E + 06	2.51E + 03	2.19E + 03	3.61E + 03	1.66E + 04	1.42E + 04
F30	Ave	3.12E + 00	2.38E + 07	6.50E + 02	1.28E + 03	1.24E + 03	3.28E + 04	8.27E + 03
	Std	2.40E + 01	1.38E + 07	3.47E − 01	2.17E − 01	4.32E − 01	5.12E − 01	6.93E − 01

reduction techniques such as principal component analysis (PCA) and linear discriminant analysis (LDA) can be used to mitigate the effects of high-dimensional problems. Additionally, in order to broaden the applicability of the algorithm, those improvements can be combined with other optimization algorithms, such as the particle swarm optimization (PSO) algorithm and the ant colony optimization (ACO) algorithm.

6. Conclusion and Future Works

This paper presents CMAIS-WOA to address the drawbacks of WOA for solving optimization problems. The proposed algorithm is based on WOA and embedded with chaotic mapping, adaptive iteration strategy, and nonlinear convergence factor. The population distribution and fitness scenarios are integrated into the optimal process by CMAIS-WOA. It enables a broad search for the best solutions. It also has the advantage of having clear steps. In the maritime industry, CMAIS-WOA is crucial for directing the design of ship bow profiles. In this paper, the effectiveness of CMAIS-WOA in solving the benchmark problem is evaluated by comparing it with other optimal algorithms. The results show that CMAIS-WOA has better optimization performance. It can effectively improve the stability of the optimal solution and help the algorithm converge to the global optimal solution.

In the future, the number of benchmark functions involved in the validation should be increased. This can further explore the law of numerical variation. In addition, CMAIS-WOA can further integrate some optimal algorithms to get hybrid algorithms with better performance, such as GWO and BA. Meanwhile, CMAIS-WOA can be applied to image processing, data mining, and other fields to solve various practical problems.

Data Availability

All data generated or analysed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Natural Science Foundation of Shandong Province (No. ZR2020MF048).

References

[1] H. R. R. Zaman and F. S. Gharehchopogh, "An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems," *Engineering Computer-Germany*, vol. 38, no. 4, pp. 2797–2831, 2022.

[2] C. Liu, "Optimal design of high-rise building wiring based on ant colony optimization," *Cluster Computing*, vol. 22, no. S2, pp. 3479–3486, 2019.

[3] E. M. Abdelkader, O. Moselhi, M. Marzouk, and T. Zayed, "An exponential chaotic differential evolution algorithm for optimizing bridge maintenance plans," *Automation in Construction*, vol. 134, no. 1, pp. 104107–104137, 2022.

[4] K. Rahul and R. K. Banyal, “Firefly algorithm: an optimization solution in big data processing for the healthcare and engineering sector,” *International Journal of Speech Technology*, vol. 24, no. 3, pp. 581–592, 2021.

[5] Y. Ye, X. Zhao, and L. Xiong, “An improved bat algorithm with velocity weight and curve decreasing,” *The Journal of Supercomputing*, vol. 78, no. 10, pp. 12461–12475, 2022.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, no. 1, pp. 46–61, 2014.

[7] A. A. Nagra, F. Han, Q. Ling, and S. Mehta, “An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization,” *IEEE Access*, vol. 7, pp. 50388–50399, 2019.

[8] H. Zhao, C. Zhang, and J. Ning, “A best firework updating information guided adaptive fireworks algorithm,” *Neural Computing and Applications*, vol. 31, no. 1, pp. 79–99, 2019.

[9] M. Wang and G. Lu, “A modified sine cosine algorithm for solving optimization problems,” *IEEE Access*, vol. 9, pp. 27434–27450, 2021.

[10] R. Salgotra and U. Singh, “The naked mole-rat algorithm,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 8837–8857, 2019.

[11] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: a new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[12] F. Soleimanian Gharehchopogh, B. Abdollahzadeh, and B. Arasteh, “An improved farmland fertility algorithm with hyper-heuristic approach for solving travelling salesman problem,” *Computer Modeling in Engineering and Sciences*, vol. 135, no. 3, pp. 1981–2006, 2023.

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

[14] S. T. Shishavan and F. S. Gharehchopogh, “An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks,” *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 25205–25231, 2022.

[15] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, and B. Abdollahzadeh, “Advances in sparrow search algorithm: a comprehensive survey,” *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 427–455, 2023.

[16] M. Samadi Bonab, A. Ghaffari, F. Soleimanian Gharehchopogh, and P. Alemi, “A wrapper-based feature selection for improving performance of intrusion detection systems,” *International Journal of Communication Systems*, vol. 33, no. 12, Article ID e4434, 2020.

[17] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, “African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems,” *Computers & Industrial Engineering*, vol. 158, Article ID 107408, 2021.

[18] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, “Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems,” *Advances in Engineering Software*, vol. 174, Article ID 103282, 2022.

[19] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, and S. Mirjalili, “Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems,” *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5887–5958, 2021.

[20] Y. Xiao, X. Sun, Y. Guo, S. Li, Y. Zhang, and Y. Wang, “An improved gorilla troops optimizer based on lens opposition-based learning and adaptive β-hill climbing for global optimization,” *Computer Modeling in Engineering and Sciences*, vol. 131, no. 2, pp. 815–850, 2022.

[21] Y. Xiao, Y. Guo, H. Cui, Y. Wang, J. Li, and Y. Zhang, “IHAOAVOA: an improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems,” *Mathematical Biosciences and Engineering*, vol. 19, no. 11, pp. 10963–11017, 2022.

[22] Y. Xiao, X. Sun, Y. Guo et al., “An enhanced honey badger algorithm based on Lévy flight and refraction opposition-based learning for engineering design problems,” *Journal of Intelligent and Fuzzy Systems*, vol. 43, no. 4, pp. 4517–4540, 2022.

[23] M. Toloueiashtian, M. Golsorkhtabaramiri, and S. Rad, “An improved whale optimization algorithm solving the point coverage problem in wireless sensor networks,” *Telecommunication Systems*, vol. 79, no. 3, pp. 417–436, 2022.

[24] D. B. Prakash and C. Lakshminarayana, “Optimal siting of capacitors in radial distribution network using whale optimization algorithm,” *Alexandria Engineering Journal*, vol. 56, no. 4, pp. 499–509, 2017.

[25] H. Zhao, S. Guo, and H. Zhao, “Energy-related CO2 emissions forecasting using an improved LSSVM model optimized by whale optimization algorithm,” *Energies*, vol. 10, no. 7, pp. 874–888, 2017.

[26] R. K. Saidala and N. Devarakonda, “Improved whale optimization algorithm case study: clinical data of anaemic pregnant woman,” *Data Engineering and Intelligent Computing*, vol. 1, pp. 271–281, 2016.

[27] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, “Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation,” *Expert Systems with Applications*, vol. 83, no. 1, pp. 242–256, 2017.

[28] K. B. O. Medani, S. Sayah, and A. Bekrar, “Whale optimization algorithm based optimal reactive power dispatch: a case study of the algerian power system,” *Electric Power Systems Research*, vol. 163, no. 1, pp. 696–705, 2018.

[29] F. S. Gharehchopogh and H. Gholizadeh, “A comprehensive survey: whale optimization algorithm and its applications,” *Swarm and Evolutionary Computation*, vol. 48, pp. 1–24, 2019.

[30] R. Salgotra, U. Singh, and S. Saha, “On some improved versions of whale optimization algorithm,” *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9653–9691, 2019.

[31] J. Zhang and J. S. Wang, “Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator,” *IEEE Access*, vol. 8, pp. 77013–77048, 2020.

[32] F. Jiang, L. Wang, and L. Bai, “An improved whale algorithm and its application in truss optimization,” *Journal of Bionic Engineering*, vol. 18, no. 3, pp. 721–732, 2021.

[33] Q. Fan, Z. Chen, Z. Li, Z. Xia, and D. Wang, “A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems,” *Engineering Computer-Germany*, vol. 37, pp. 1851–1878, 2021.

[34] C. Tang, W. Sun, M. Xue, X. Zhang, H. Tang, and W. Wu, “A hybrid whale optimization algorithm with artificial bee colony,” *Soft Computing*, vol. 26, no. 5, pp. 2075–2097, 2022.

[35] M. Li, G. Xu, Q. Lai, and J. Chen, “A chaotic strategy-based quadratic Opposition-Based Learning adaptive variable-speed whale optimization algorithm,” *Mathematics and Computers in Simulation*, vol. 193, pp. 71–99, 2022.

[36] D. Oliv, A. E. A. Mohamed, and A. E. Hassanien, “Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm,” *Applied Energy*, vol. 1, pp. 141–154, 2017.

[37] F. S. Gharehchopogh, “Quantum-inspired metaheuristic algorithms: comprehensive survey and classification,” *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5479–5543, 2023.

[38] M. Li, X. Yu, B. Fu, and X. Wang, “A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems,” *Neural Computing and Applications*, pp. 1–14, 2023.

[39] M. S. Uzer and O. Inan, “Application of improved hybrid whale optimization algorithm to optimization problems,” *Neural Computing and Applications*, vol. 35, no. 17, pp. 12433–12451, 2023.

[40] Q. Fan, Z. Chen, W. Zhang, and X. Fang, “ESSAWOA: enhanced whale optimization algorithm integrated with salp swarm algorithm for global optimization,” *Engineering with Computers*, vol. 38, no. S1, pp. 797–814, 2022.

[41] R. Haupt and S. Haupt, *Practical Genetic Algorithm*, John Wiley and Sons, Hoboken, NJ, USA, 2004.

[42] G. Y. Ning and D. Q. Cao, “Improved whale optimization algorithm for solving constrained optimization problems,” *Discrete Dynamics in Nature and Society*, vol. 2021, no. 12, Article ID 8832251, 13 pages, 2021.

[43] M. Ali and M. Pant, “Improving the performance of differential evolution algorithm using Cauchy mutation,” *Soft Computing*, vol. 15, no. 5, pp. 991–1007, 2011.

[44] R. Salgotra, U. Singh, S. Singh, and N. Mittal, “A hybridized multi-algorithm strategy for engineering optimization problems,” *Knowledge-Based Systems*, vol. 217, Article ID 106790, 2021.