Article

An Improved Whale Optimization Algorithm with Random Evolution and Special Reinforcement Dual-Operation Strategy Collaboration

Qibing Jin, Zhonghua Xu * and Wu Cai

Institute of Automation, Beijing University of Chemical Technology, Beijing 100029, China; jinqb@mail.buct.edu.cn (Q.J.); buctcaiwu@163.com (W.C.)
* Correspondence: xzhbuct@163.com

Abstract: In view of the slow convergence speed, difficulty of escaping from the local optimum, and difficulty maintaining the stability associated with the basic whale optimization algorithm (WOA), an improved WOA algorithm (REWOA) is proposed based on dual-operation strategy collaboration. Firstly, different evolutionary strategies are integrated into different dimensions of the algorithm structure to improve the convergence accuracy and the randomization operation of the random Gaussian distribution is used to increase the diversity of the population. Secondly, special reinforcements are made to the process involving whales searching for prey to enhance their exclusive exploration or exploitation capabilities, and a new skip step factor is proposed to enhance the optimizer’s ability to escape the local optimum. Finally, an adaptive weight factor is added to improve the stability of the algorithm and maintain a balance between exploration and exploitation. The effectiveness and feasibility of the proposed REWOA are verified with the benchmark functions and different experiments related to the identification of the Hammerstein model.

Keywords: computation intelligence; whale optimization algorithm; Hammerstein model; function optimization; system identification; swarm intelligence

1. Introduction

There are many optimization problems across various fields in the real world. The characteristics of these types of problems are that an optimal solution or parameter value must be found among many solutions or parameter values under certain conditions, with the mathematical calculation method required to solve such problems being the optimization algorithm. The traditional optimization algorithms included hill-climbing [1], random search [2], and Newton’s method [3], which are only suitable for solving small-scale problems, not for complex high-dimensional, nonlinear, and multimodal optimization problems [4,5]. Therefore, researchers need to find new methods to solve complex optimization problems. In recent decades, meta-heuristic optimization algorithms have attracted widespread attention from researchers. Inspired by human intelligence, the social nature of biological groups, and the laws of natural phenomena, many meta-heuristic optimization algorithms [6] have been proposed to solve these problems. Compared with traditional optimization algorithms, meta-heuristic optimization algorithms have fewer control parameters, greater randomness, stronger adaptability, and do not require organizational structure information for the algorithm. Hence, meta-heuristic algorithms can obtain better optimization results and are widely used in various optimization fields [7].

With in-depth research in the field of optimization, some well-known meta-heuristic algorithms have been proposed, such as the genetic algorithm (GA) [8], biogeography-based optimization (BBO) [9], differential evolution (DE) [10], particle swarm optimization (PSO) [11], ant colony optimization (ACO) [12], artificial bee colony (ABC) [13], krill herd (KH) [14], monarch butterfly optimization (MBO) [15], earthworm optimization algo-
rithm (EWA) [16], elephant herding optimization (EHO) [17], Harris hawks optimization (HHO) [18], slime mold algorithm (SMA) [19], moth search algorithm (MSA) [20], whale optimization algorithm (WOA) [21], and salp swarm algorithm (SSA) [22] techniques. These meta-heuristic algorithms have their own characteristics and are used to solve various optimization problems. For example, the performance of evolution-based algorithms depends on the choice of mutation strategies and control parameters, with different mutation strategies and control parameters required for different optimization problems are [23,24]. The population-based meta-heuristic algorithms generally include two search processes: exploration (diversification) and exploitation (intensification) [25–27]. In the exploration phase, the optimizer explores in a limited solution space, while the exploration step must be random enough to ensure that it can extend everywhere in the entire search space [28].

The exploitation phase is a local development process that occurs after the exploration operation, which represents the local search capability of the algorithm. In this phase, the optimizer only focuses on the neighborhood of the high-quality solutions rather than the entire solution space. To enhance the performance of algorithms, it is essential to keep an appropriate balance between exploration and exploitation [29,30]. With the development of technology and the broadening of the scope of engineering problems, it is important for researchers to continuously study and design novel optimization methods to solve actual application problems. The hybrid optimization strategy is one of the most effective ways to design new optimization algorithms [31], which can make various single algorithms complement each other, resulting in better optimization efficiency. For example, Li et al. [32] proposed a hybrid optimization strategy to identify the parameters of nonlinear systems, incorporating the escape characteristics of simulated annealing (SA) [33] into PSO. Mo [34] introduced a new hybrid optimization algorithm called a particle-swarm-assisted incremental evolution strategy (PIES) for function optimization problems. Ghanem et al. [35] combined parts of the artificial bee colony (ABC) with elements from the monarch butterfly optimization (MBO) approach to improve the performance of solving numerical optimization problems. Javaid et al. [36] proposed a hybrid meta-heuristic technique for management of home energy, which integrated enhanced differential evolution (EDE) and earthworm optimization algorithm (EWA) techniques.

In this paper, we focus on the whale optimization algorithm. The WOA was first proposed in 2016, and is a meta-heuristic optimization algorithm that simulates the hunting behavior of humpback whales [21]. Compared with other meta-heuristic algorithms, the WOA is comparatively easy to implement and has fewer control parameters. Due to its simple structure and ease of implementation, the WOA has also been widely applied in various fields and disciplines, including engineering [37–42]. However, the WOA inextricably has some inherent disadvantages common to meta-heuristics, such as a tendency toward local optimality, slow convergence, and low accuracy [43,44]. Therefore, some WOA variants have been proposed to overcome the shortcomings of the basic WOA. Mohammedardi et al. [43] proposed a new hybrid algorithm (WOABAT) by using the bat algorithm (BAT) to replace the exploration phase of WOA to solve function optimization problems. Amarjeet et al. [45] hybridized the WOA with a Laplace crossover operator to overcome the premature convergence and stagnation problems of the pure WOA. Gehad et al. [46] presented a novel chaotic whale optimization algorithm (CWOA), which embeds a chaotic search into the WOA search iterations. Fan et al. [47] proposed a novel hybrid algorithm that combines basic the WOA and SSA to solve global optimization problems. Yan et al. [48] proposed a hybrid whale optimization algorithm based on the Lévy flight strategy and lateral inhibition to solve an underwater image matching problem. Rohit et al. [49] introduced three different modified WOA algorithms to improve its ability for exploration, which were based on opposition-based learning, rapidly decreasing parameters, and initializing the worst particles. Sally et al. [50] integrated crossover and mutation operations with the WOA to optimize the spectrum utilization. Majdi et al. [51] proposed a hybridization model (WOASAT) based on the WOA and SA to solve different feature selection problems.
During our previous research, we found that WOA variants generally come in only two forms. One involves the complement of two different heuristic algorithms [47], while the other one involves the replacement of the whale updating location strategy with the evolutionary operations of the evolution-based algorithm [50]. Although the performance of the algorithm is improved to some extent, there are still some shortcomings. For example, the simple hybrid of two algorithms will cause the population information disorder to lose evolutionary direction and fall into a local optimum. For the above point, we creatively propose an enhanced whale optimization algorithm combining the two improvement methods, called the random evolutionary whale optimization algorithm (REWOA). The main work of the REWOA is divided into two parts: structure optimization and location updating. The enhanced evolutionary operations are integrated with different dimensions of the WOA structure to optimize the algorithm operation structure, and randomization operations are added to improve the exploration capability. Inspired by the Lévy flight strategy [52], we created a step factor with stronger escape ability to improve the ability to escape the local optimum. Professional enhancements were made to the original WOA location updating mechanism; that is, relevant modifications were made to the characteristics of the three hunting behaviors of the WOA. These improvements enhanced the performance of the novel optimizer and improved the search accuracy. In order to balance diversification and intensification, the weight factor was added to the bubble net attacking method to divide the hunting activity into the two stages.

To verify the performance of the REWOA, we tested it through 23 different benchmark functions and three Hammerstein model identification problems. The results obtained were analyzed and compared with the WOA and its variants, as well as with other well-known meta-heuristics. The comparative results showed that REWOA can provide faster and more accurate convergence results.

The rest of this paper is organized as follows. Section 2 describes the structure and principles of the basic WOA in detail. An improved WOA algorithm (REWOA) is proposed in Section 3. The proposed algorithm is evaluated with 23 different benchmark functions and three Hammerstein model identification problems in Sections 4 and 5, respectively. Finally, concluding remarks are given in Section 6.

2. Whale Optimization Algorithm (WOA)

The whale optimization algorithm was first proposed in 2016 and is a meta-heuristic optimization algorithm based on simulating the hunting behavior of humpback whales [21]. Based on the hunting behavior of humpback whales, the WOA algorithm could be described abstractly as involving three types of predation strategies, including encircling prey, the bubble net attacking method, and searching for prey. The mathematical model is described in this subsection.

2.1. Encircling Prey

Each humpback whale represents a search agent, and their positions in the search space represent solutions to problems. The candidate optimal solution is the position of the prey and its neighborhood in the search space. The position updating mechanism of the population revolves around the position of the current optimal candidate solution. The mathematical equations for this behavior are shown below:

\[
\vec{D} = \left| \vec{C} \cdot \vec{X}^* (t) - \vec{X}(t) \right| \tag{1}
\]

\[
\vec{X}(t + 1) = \vec{X}^* (t) - \vec{A} \cdot \vec{D} \tag{2}
\]
where \( t \) indicates the current iteration, \( X \) and \( X^* \) represent position vectors with the current iteration and the location information of the best solution so far, respectively; \( A \) and \( C \) are coefficient vectors given by Equations (3) and (4), respectively.

\[
\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{3}
\]
\[
\vec{C} = 2 \cdot \vec{r} \tag{4}
\]
\[
\vec{a} = 2 - 2 \cdot t / t_{\text{max}} \tag{5}
\]

where \( a \) is the convergence factor, which decreases linearly from 2 to 0 with the increase of the number of iterations, and \( r \) is the random vector in \([0, 1]\).

### 2.2. Bubble Net Attacking Method

The bubble net attack hunting behavior can be divided into two parts. The first is the shrinking and encircling of the prey by the whale, and the second is the spiral upward encirclement and suppression. In view of the above description, the two location updating mechanisms of the WOA can be obtained by mathematical modeling. The shrinking and encircling mechanism is achieved by continuously reducing the value of \( \vec{a} \). When \(|A| \leq 1\), the new position of the search agent is updated according to Equation (2). The spiral updating position mimics the behavior of a humpback whale approaching its prey in a spiral motion. The mathematical model is as follows:

\[
\vec{X}(t + 1) = D' \cdot e^{bl} \cdot \cos(2 \cdot \pi \cdot l) + \vec{X}^*(t) \tag{6}
\]

where \( D' = |X^* - X| \), which indicates the distance of each agent to the current optimal solution; \( b \) is a constant used to define the shape of the logarithmic spiral and \( l \) is a random number in \([-1, 1]\). In the real world, a whale hunts its prey by using both methods at the same time. To simulate this simultaneous behavior, WOA sets the same probability to choose the above two different search mechanisms to update the positions of the whales during optimization, which could be defined as Equation (7):

\[
\vec{X}(t + 1) = \begin{cases} 
\vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\
D' \cdot e^{bl} \cdot \cos(2 \cdot \pi \cdot l) + \vec{X}^*(t) & \text{if } p \geq 0.5 
\end{cases} \tag{7}
\]

### 2.3. Searching for Prey

In the meta-heuristic optimization algorithm, it is important to balance the exploitation stage and exploration stage in the optimization problem. In the WOA, the balance ability for exploration and exploitation can be manifested through the size of vector \( A \). When \( A \geq |1| \), the whales update their positions by searching for the prey towards the random search agent to determine the global optimum and eliminate many local minima. The mathematical model is formulated as:

\[
\vec{D} = \left| \vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X} \right| \tag{8}
\]

\[
\vec{X}(t + 1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D} \tag{9}
\]

where \( X_{\text{rand}} \) represents the random vector. The flow chart of the pure WOA is illustrated in Figure 1.
3. Random Evolutionary Whale Optimization Algorithm (REWOA)

The unique structure of the WOA means it has the ability to balance exploration and exploitation, however this causes the negative impacts of poor convergence accuracy and slow convergence speed. Here, we propose an improved whale optimization algorithm. The main work is divided into two parts—structural optimization and transformation and strengthening of the location updating mechanism. The first part involves the integration of random evolution strategies into the unique structure of the WOA. The purpose of this approach is to not only retain the balance ability but to also improve the convergence accuracy and speed of the algorithm. The next is a special reinforcement of the WOA location updating mechanism. Simply put, it improves the ability of whales to hunt prey and enhances the overall performance of the algorithm. In this subsection, the REWOA will be presented in detail.

3.1. Random Evolutionary

In the population-based optimization algorithm, to a certain extent it is difficult to balance the global exploration and the local exploitation. The evolutionary strategy has the characteristics of guidance, parallelism, and randomness, which can ensure the stability of the algorithm. Therefore, the evolutionary strategy is introduced to improve the balance capability of the algorithm. Firstly, the optimizer selects some of the whales for

---

Initialize the population of whales $X_i$ ($i = 1, 2, 3 \cdots n$)
Calculate the fitness of each search agent
$X^*$ represents the best search agent

while ($t < \text{maximum number of iterations}$)
for each search agent
Update the control parameters
if $1(p < 0.5)$
if $2(|A| < 1)$
Update the position of the current whale using $\bar{X}(t+1) = \bar{X}^*(t) - \bar{A} \cdot \bar{D}$
elseif $2(|A| \geq 1)$
Select a random whale $X_{\text{rand}}$
Update the position of the current whale using $\bar{X}(t+1) = X_{\text{rand}} - \bar{A} \cdot \bar{D}$
end if
else if $1(p \geq 0.5)$
Update the position of the current whale using $\bar{X}(t+1) = \bar{D} \cdot e^t \cdot \cos(2 \cdot \pi \cdot t) + \bar{X}^*(t)$
end if
end for
Check if any whale goes beyond the search space and amend it
Calculate the fitness of each novel whale
Update $X^*$ if there is a better solution
t = t + 1
end while
return $X^*$

---

Figure 1. Pseudo-code of the WOA algorithm.
differential mutation with a certain probability and adds random Gaussian interference to improve the randomness of the whale population distribution and increase the global search ability. Then, crossover and selection strategies are integrated into the structure of the optimizer to ensure the correctness of the population evolution direction and to speed up the convergence. In the basic WOA algorithm, due to the single information communication method and the concentrated population distribution of whales, the global search ability of whales is poor. To optimize the population distribution and improve the global search ability of the algorithm, we set a mutation probability of 0.2 based on Rechenberg criterion to randomly select whales to perform differential mutation operations. This operation is implemented using two different information exchange channels and by applying random Gaussian disturbances. The mathematical model of the mutation operation is provided as follows:

\[
\vec{D}^* = \begin{bmatrix} \vec{X}^*(t) - \vec{X}(t) \end{bmatrix}
\]

\[
\vec{D}_r = \begin{bmatrix} \vec{X}_{r1}(t) - \vec{X}_{r2}(t) + \vec{X}_{r3}(t) - \vec{X}_{r4}(t) \end{bmatrix}
\]

\[
\vec{X}(t + 1) = \vec{X}(t) + r \cdot \vec{D}^* + G \cdot \vec{D}_r + r(0, 1) \cdot G
\]

where \(\vec{X}_{r1}, \vec{X}_{r2}, \vec{X}_{r3}, \) and \(\vec{X}_{r4}\) represent four different individuals selected at random in the population, except the current one. The information exchange vector between the current individual and the optimal individual is \(\vec{D}^*\). The information exchange channel for four different individuals selected at random is \(\vec{D}_r\). \(G\) is the Gaussian distribution, the mean value and variance of which are 0 and 1, while \(r(0, 1)\) indicates the Boolean variable.

To increase the potential diversity of the population, we set a crossover probability to determine whether to perform a location update operation. The crossover operation can be summarized as Equation (13).

\[
X^i_j(t + 1) = \begin{cases} 
X^i_j(t) & \text{if } r < cr \\
X^i_j(t + 1) & \text{if } r \geq cr 
\end{cases}
\]

where \(X^i_j(t)\) represents the value of the \(j\)-th dimension of the current \(i\)-th individual; \(cr\) indicates the crossover probability, which mainly reflects the running time and convergence speed of the algorithm. The smaller the value of \(cr\), the longer the running time, the faster the convergence speed, and the smaller the population diversity. To overcome this shortcoming, we updated the value of \(cr\) using Equation (14):

\[
cr = m + (0.5 - m) \cdot \sin(t \cdot \pi / 2 \cdot t_{\text{max}})
\]

where \(t_{\text{max}}\) represents the maximum number of iterations and \(m\) is a constant in the range of \([0, 0.5]\) to control the fluctuation of \(cr\). This modified \(cr\) ensures the balance between the diversity and reinforcement of the whale’s position.

The new whale obtained after differential mutation and crossover operation will be compared with the original whale for fitness, and the individual with better fitness will be selected as the next-generation individual to participate in the next iteration. This process can be described as follows:

\[
\vec{X}(t + 1) = \begin{cases} 
\vec{X}(t + 1) & \text{if } f(\vec{X}(t + 1)) \leq f(\vec{X}(t)) \\
\vec{X}(t) & \text{otherwise}
\end{cases}
\]

According to the above formula, the selection process ensures that all individuals in the population will improve.
3.2. Special Reinforcement

The traditional WOA has slow convergence speed and easily falls into the local optimal solution. To overcome these shortcomings, we proposed certain improvement measures for the three types of WOA updating mechanisms. For the steps involving encircling prey and searching for prey, the basic idea of the improvement measures is to separately enhance the capabilities of exploration and exploitation and to set up a mechanism to escape the local optimal solution. For the exploration stage, the current individual communicates with the random individual and exponentially expands the distance of exploration. Then, the optimizer guides the evolution of the population through the elite individual. These measures are shown as the following equations:

\[ D_r^j = \left| 2 \cdot G^j \cdot X_{\text{rand}}^j - X_i^j \right| \tag{16} \]

\[ D_e^j = RW^j \cdot \left| X^* - X_i^j \right| \tag{17} \]

\[ X_i^j(t + 1) = \begin{cases} 
X_{\text{rand}}^j(t) - A^j \cdot D_r^j & \text{if } |A^j| > 1 \\
X^*(t) - A^j \cdot D_e^j & \text{if } |A^j| \leq 1 
\end{cases} \tag{18} \]

where \( X_i^j \), \( X^j \), and \( X_{\text{rand}}^j \) are the values of the \( j \)-th dimension of the current, optimal, and randomly selected individuals, respectively. The difference from the pure WOA is that the location information for each whale is jointly determined using two improved mechanisms for encircling and searching for prey. In the beginning stages of the basic WOA, as the individual differences in the population are relatively large, the optimizer has strong global search capabilities. However, with the continuous iteration of the algorithm, the gradual convergence of individuals in the population causes its search ability to be weakened, so that it is difficult to escape the local optimum. Therefore, we dynamically adjusted and improved the convergence factor to improve the optimization performance. The basic WOA convergence factor linearly decreases from 2 to 0 with the increase in the number of iterations. This causes the parameter \( A \) to infinitely tend to 0, which makes it difficult to update individual location information and limit the search ability. Regarding the inspiration for the improvement strategy in [53], the control parameter \( A \) can be expressed by Equation (19) to balance the searching ability of the algorithm, thereby improving the overall optimization performance of the algorithm. Inspired by the Levy flight strategy [52], we introduced a step factor \( RW \), improving the randomization of the algorithm to improve the vitality of individual whales so as to escape the local optimum. Randomization plays an important role in exploration, exploitation, and diversification. Spatial complexity depends on spatial randomness, but it is difficult to assess spatial randomness in terms of algorithmic complexity [54]. Therefore, we implemented an experiment [55] to assess spatial randomness, using Gaussian step-size distribution, Levy flight, and RW step-size distribution methods to randomly walk through 100 consecutive steps, starting from the origin (0, 0). The results are shown in Figure 2. The generation between short and long steps is more random, which maintains the balance between local and out-of-local searches. \( RW \) is more likely to involve a big step and have the greatest escape power.

\[ A^j = a \cdot \left( (1 - G^j) / (2 \cdot (1 + 2 \cdot r^j)) \right) \tag{19} \]

\[ RW^j = \tan(\pi \cdot (r^j - 0.5)) / (4 \cdot (1 + 0.5 \cdot G^j)) \tag{20} \]
where \( a = 2 - (t/t_{\text{max}}) \) decreases linearly from 2 to 1. For the bubble net attacking method, we adopted the dynamic inertia weight \( \omega \) to balance the local and global searching. The mathematical formula can be expressed as:

\[
\vec{X}(t + 1) = D' \cdot e^{lt} \cdot \cos(2 \cdot \pi \cdot l) + \omega \cdot \vec{X}^*(t)
\]

(21)

\[
\omega = 1 - \left( \frac{t}{t_{\text{max}}} \right)^{1/t}
\]

(22)

\[
pr = 1.2 - 0.9 \cdot \cos(t \cdot \pi / t_{\text{max}})
\]

(23)

where \( \omega \) is a random value decreasing from 1 to 0 with the continuous iteration of the algorithm. This will cause \( \omega \) to infinitely tend to 0, meaning it will lose its optimal whale information and causing the evolution of the population to lose its navigation. Therefore, we set an adaptive probability \( pr \) given by Equation (23) to ensure that sufficient information about elite whales guided the population evolution at all times.

Figure 2. Random walk through of 100 consecutive steps, starting at the origin (0, 0): (a) Levy flight step-size distribution; (b) RW step-size distribution; (c) Gaussian step-size distribution.

3.3. Main Procedure of the REWOA

The REWOA randomly combines different evolution operations with the pure WOA, and special enhancements were made to the exploration and exploitation strategies of the whale algorithm. The main procedure of the REWOA is summarized in Figure 3.

3.4. Complexity Analysis

In the section, the computational complexity of the REWOA is analyzed to reflect the operating efficiency of the algorithm. The computational complexity of the algorithm is divided into time complexity and space complexity. Space complexity indicates the amount of storage space required by the algorithm. The space complexity of the algorithm is only related to the population number and the dimensions of the optimization problem.

The REWOA and WOA numbers of the population are both \( N \), while the dimensions of the optimization problem are represented as \( D \). Therefore, the total space complexity of the REWOA is \( O(N \ast D) \), which is the same as that of the WOA.

The REWOA consists of three major processes: initialization, the main loop, and halting judgment. The time complexity analysis for the REWOA was developed for the three different processes of the algorithm. In the REWOA, \( Miter \) is the maximum number of iterations; \( N \) is the population number; \( D \) indicates the dimensions of the problem; \( t \) is the time taken to update the position of each dimension of each agent; \( f(N) \) is the time taken to calculate the fitness value.
Start

Initialize the whale population

Calculate the fitness of each whale, determine the best whale and optimal solution

iterations< maximum

YES

End

Update Control parameters

YES

NO

If rand< cr

Update the whale position using Equation (25)

NO

If |A|<1

Update the whale position using Equation (25)

NO

If rand< 0.2

Update the whale position using Equation (19)

NO

Calculate the fitness of whale and greedy selection using Equation (22)

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

Calculate the fitness of whale and greedy selection using Equation (22)

Update X* if there is a better solution

Iteration=iteration+1

Figure 3. Flowchart for the REWOA. where \( cr \) indicates the crossover probability using Equation (14), \( pr \) indicates adaptive probability using Equation (23).
In the initialization, the time complexity of the REWOA is \( T_1 = O(N(D \cdot t)) = O(N \cdot D) \), which is the same as that of the WOA. In the main loop, \( t_1, t_2, t_3, \) and \( t_4 \) are the times taken to update parameter, perform greedy selection, check if the whale goes beyond the search space, and update the optimal solution, respectively. In this stage, the time complexity includes the following three parts:

\[
T_{21} = \frac{1}{3} (O(N(D \cdot t) + f(N)))
\]

\[
T_{22} = \frac{2}{3} (O(N((1 - cr) \cdot D \cdot t) + f(N)))
\]

\[
T_{23} = O(t_1 + t_2 + t_3 + t_4)
\]

The time complexity of the main loop stage is the sum of the above three parts, as shown in Equation (27).

\[
T_2 = \text{Miter}(T_{21} + T_{22} + T_{23})
\]

With this, the time complexity can be calculated as \( T_2 = O(N \cdot D + f) \). The time complexity of the main loop of the WOA is indicated by Equation (30). The time complexity of this stage of the REWOA and WOA is the same.

\[
T_{21}' = O(N(D \cdot t) + f(N))
\]

\[
T_{22}' = O(t_1' + t_3 + t_4)
\]

\[
T_2' = \text{Miter}(T_{21}' + T_{22}') = O(N \cdot D + f)
\]

where \( T_{21}' \) and \( T_{22}' \) are the time complexity values of the two main parts of the WOA, respectively; \( t_1' \) is the time taken to update the parameters of the WOA. The time complexity values of the last steps of the REWOA and WOA are both \( T_3 = O(1) \).

Based on the above analysis, we can conclude that the proposed REWOA does not reduce the execution efficiency of the algorithm.

4. Experimental Results and Discussion

In order to investigate the numerical efficiency of the proposed REWOA, the 23 classical benchmark functions from [21] were utilized to check the performance of the algorithm, as shown in Table 1. These functions are divided into three groups: unimodal (F1–F7), multimodal (F8–F13), and fixed-dimension multimodal (F14–F23). The unimodal benchmark functions with a only single optimal solution can be utilized to verify the exploitation and convergence. There are many optimal solutions for the multimodal benchmark functions. It is worth mentioning that among the many optimal solutions, most are local optima, and there is only one global optimum. The fixed dimensional multimodal functions have the ability to define the desired number of design variables and could provide a different search space. Therefore, the multimodal functions are responsible for testing exploration and avoiding the entrapment in the local optimal solution. In Table 1, the corresponding properties of these functions are listed, where dim represents the dimensions of the functions and range indicates the scope of the search space.
Table 1. Description of benchmark functions.

| No. | Formula | Dim | Range | \( f_{\text{min}} \) |
|-----|---------|-----|-------|---------------|
| F1  | \( f(x) = \sum_{i=1}^{n} x_i^2 \) | 30  | \([-100, 100]\) | 0             |
| F2  | \( f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \) | 30  | \([-10, 10]\) | 0             |
| F3  | \( f(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} x_j \right)^2 \) | 30  | \([-100, 100]\) | 0             |
| F4  | \( f(x) = \max \{ |x_i|, 1 \leq i \leq n \} \) | 30  | \([-100, 100]\) | 0             |
| F5  | \( f(x) = \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \) | 30  | \([-30, 30]\) | 0             |
| F6  | \( f(x) = \sum_{i=1}^{n-1} \left( x_i + 0.5 \right)^2 \) | 30  | \([-100, 100]\) | 0             |
| F7  | \( f(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1] \) | 30  | \([-1.28, 1.28]\) | 0             |
| F8  | \( f(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \) | 30  | \([-500, 500]\) | -12,569.49   |
| F9  | \( f(x) = \sum_{i=1}^{n} \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right) \) | 30  | \([-5.12, 5.12]\) | 0             |
| F10 | \( f(x) = \sum_{i=1}^{n} \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right) \) | 30  | \([-32, 32]\) | 0             |
| F11 | \( f(x) = \frac{1}{100} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{\pi x_i}{\sqrt{10}}) + 1 \) | 30  | \([-600, 600]\) | 0             |
| F12 | \( f(x) = \frac{1}{10} \left( 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} \left( (y_i - 1)^2 + 10 \sin^2(\pi y_i) + (y_i - 1)^2 \right) \right) + \sum_{i=1}^{n} u(x_i, 10, 100, 4) \) | 30  | \([-50, 50]\) | 0             |
| F13 | \( y_i = 1 + \frac{1}{8} \left( x_i a, k, m \right) = \begin{cases} \frac{k_1}{x_i - a}, & x_i > a \\ 0, & -a < x_i < a \\ \frac{k_1}{x_i - a}, & x_i < a \end{cases} \) | 30  | \([-50, 50]\) | 0             |
| F14 | \( f(x) = \left( \frac{1}{\cos^3 1} + \sum_{i=1}^{n} \frac{1}{1 + \cos(2\pi x_i)} \right)^{-1} \) | 2   | \([-65, 65]\) | 1             |
| F15 | \( f(x) = \sum_{i=1}^{n} \left[ |a_1 - x_i| \frac{|x_1 + x_2|}{|x_1 + x_2|} \right]^2 \) | 4   | \([-5, 5]\) | 0.0003        |
| F16 | \( f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \) | 2   | \([-5, 5]\) | -1.0316       |
| F17 | \( f(x) = \frac{1}{2} \left( x_1^2 + \frac{1}{x_1} - 6 \right)^2 + \frac{1}{10} \left( 1 - \frac{1}{x_1} \right) \cos x_1 + x_1 + 10 \) | 2   | \([-5, 5]\) | 0.398         |
| F18 | \( f(x) = \left[ 1 + \left( x_1 + x_2 + 1 \right)^2 \left( 19 - 14x_1 + 3x_1^2 + 14x_2 - 6x_1x_2 + 3x_2^2 \right) \right] \) | 2   | \([-2, 2]\) | 3             |
| F19 | \( f(x) = -\sum_{i=1}^{n} a_i \exp(-\sum_{j=1}^{n} a_j (x_j - p_{ij})^2) \) | 3   | \([1, 3]\) | -3.86         |
| F20 | \( f(x) = -\sum_{i=1}^{n} a_i \exp(-\sum_{j=1}^{n} a_j (x_j - p_{ij})^2) \) | 6   | \([0, 1]\) | -3.32         |
| F21 | \( f(x) = -\sum_{i=1}^{n} (X - a_i) (X - a_i)^T + c_i \) | 4   | \([0, 10]\) | -10.1532      |
| F22 | \( f(x) = -\sum_{i=1}^{n} (X - a_i) (X - a_i)^T + c_i \) | 4   | \([0, 10]\) | -10.4028      |
| F23 | \( f(x) = -\sum_{i=1}^{n} (X - a_i) (X - a_i)^T + c_i \) | 4   | \([0, 10]\) | -10.5363      |
In this paper, we use the average value and standard deviation of the solution to evaluate the basic performance of the proposed algorithm. The mathematical formula can be expressed as:

\[
\text{Avg} = \frac{1}{n} \sum_{i=1}^{n} S_i \tag{31}
\]

\[
\text{Std} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (S_i - \text{Avg})^2} \tag{32}
\]

where \( S_i \) indicates calculation results for the \( i \)-th time and \( n \) is the number of runs. The mean value can more fairly show the calculation results of the algorithm and avoid the particularity in the calculation process. The standard deviation can reflect the discrete degree of calculation results. These indices are used to measure the approximation degree of the algorithm to the optimal solution under the random initial value and to measure the dependence degree of the algorithm’s performance on the random initial value operation. The smaller the value, the better the robustness and reliability of the algorithm.

In the experiment, the REWOA is compared with basic metaheuristic algorithms (WOA, SSA, DE) and WOA variants (WOABAT, WOASAT) to verify the improved calculation performance. For a fair comparison, the maximum iteration number for all algorithms is set to 500 and the size of population is set to 30. For each benchmark function, all optimizers were run 30 times independently to ensure the stability of the algorithm. Table 2 shows the control parameter values for all algorithms that participated in the experiment, which were derived from the literature [10,21,22,43,51]. Then, the REWOA, WOA, WOABAT, WOASAT, SSA, and DE are tested on these problems concurrently. The statistical results are reported in Tables 3–5, where \( \text{best} \) and \( \text{sd} – \text{best} \) represent the number of optimal and suboptimal function optimization results, respectively.

**Table 2. The control parameters settings.**

| Method       | Control Parameter                        | Value       |
|--------------|------------------------------------------|-------------|
| REWOA       | Convergence factor \((a)\)                | [1, 2]      |
|             | Mutation probability \((mr)\)            | 0.2         |
|             | Crossover probability \((cr)\)           | [0, 0.5]    |
|             | Adaptive weight \((\omega)\)             | [0, 1]      |
| WOA [21]    | Convergence factor \((a)\)               | [0, 2]      |
|             | Probability coefficient \((p)\)          | 0.5         |
| WOABAT [43] | Convergence factor \((a)\)               | [0, 2]      |
|             | Probability coefficient \((p)\)          | 0.5         |
|             | Pulse rate \((r)\)                       | 0.5         |
|             | Sound loudness \((A_i)\)                 | 0.5         |
| SSA [22]    | Probability coefficient \((p)\)          | 0.5         |
| WOASAT [51] | Reduction rate \((\alpha)\)              | 0.99        |
|             | Initial temp \(t_0\)                     | 0.1         |
|             | Maximum Number of Iterations             | 30          |
|             | Convergence factor \((a)\)               | [0, 2]      |
|             | Probability coefficient \((p)\)          | 0.5         |
| DE [10]     | Mutation operator \((F)\)                | 0.5         |
|             | Crossover probability \((cr)\)           | 0.3         |
Table 3. Comparison of REWOA with other algorithms for use on unimodal problems. Note that the bold letters in the table indicate the respective best results.

| Function | Metric | REWOA     | WOA       | WOABAT    | SSA      | WOASAT    | DE        |
|----------|--------|-----------|-----------|-----------|----------|-----------|-----------|
| F1       | avg    | $4.2 \times 10^{-222}$ | $3.67 \times 10^{-73}$ | $1.57 \times 10^{-06}$ | $1.60 \times 10^{-07}$ | 0         | $3.44 \times 10^{-15}$ |
| F1       | std    | 0         | $1.12 \times 10^{-72}$ | 7.11 $\times 10^{-07}$ | 2.24 $\times 10^{-07}$ | 0         | $1.04 \times 10^{-14}$ |
| F2       | avg    | $2.60 \times 10^{-213}$ | $5.57 \times 10^{-52}$ | 7.33 $\times 10^{-03}$ | 2.34      | 0         | $2.10 \times 10^{-09}$ |
| F2       | std    | 0         | $1.82 \times 10^{-51}$ | $1.39 \times 10^{-03}$ | 1.49      | 0         | $3.82 \times 10^{-9}$  |
| F3       | avg    | 0         | $4.21 \times 10^{4}$   | $9.56 \times 10^{-06}$ | 1.53      | $7.93 \times 10^{-01}$ | $4.07 \times 10^{-5}$  |
| F3       | std    | 0         | $1.48 \times 10^{-04}$ | $2.17 \times 10^{-06}$ | $7.48 \times 10^{-2}$  | $5.00 \times 10^{-01}$ | $1.99 \times 10^{-3}$  |
| F4       | avg    | $5.65 \times 10^{-151}$ | 49.6         | 1.01 $\times 10^{-03}$ | 11.4      | 0         | $8.04 \times 10^{-01}$ | 8.30      |
| F4       | std    | 2.79 $\times 10^{-150}$ | 29.4         | $8.71 \times 10^{-05}$ | 3.93      | $3.92 \times 10^{-2}$  | 3.90      |
| F5       | avg    | 2.64      | 27.9       | 6.58      | $3.40 \times 10^{2}$   | 29.6      | 62.1      |
| F5       | std    | 7.88      | $4.88 \times 10^{-01}$ | 11.9     | $4.59 \times 10^{2}$   | 19.0      | 51.9      |
| F6       | avg    | $2.01 \times 10^{-08}$ | 3.64 $\times 10^{-01}$ | 1.71 $\times 10^{-06}$ | $2.72 \times 10^{-07}$ | 1.28 $\times 10^{-03}$ | $6.90 \times 10^{-14}$ |
| F6       | std    | 3.94 $\times 10^{-08}$ | 2.20 $\times 10^{-01}$ | $8.14 \times 10^{-07}$ | $3.02 \times 10^{-07}$ | $7.99 \times 10^{-04}$ | $1.56 \times 10^{-13}$ |
| F7       | avg    | $1.47 \times 10^{-03}$ | $2.92 \times 10^{-03}$ | $4.85 \times 10^{-04}$ | $1.74 \times 10^{-01}$ | $5.99 \times 10^{-02}$ | $2.76 \times 10^{-01}$ |
| F7       | std    | $1.26 \times 10^{-03}$ | $3.34 \times 10^{-03}$ | $8.45 \times 10^{-04}$ | $5.91 \times 10^{-02}$ | $3.59 \times 10^{-02}$ | $2.61 \times 10^{-01}$ |

Table 4. Comparison of REWOA with other algorithms for use on multimodal problems. Note that the bold letters in the table indicate the respective best results.

| Function | Metric | REWOA     | WOA       | WOABAT    | SSA      | WOASAT    | DE        |
|----------|--------|-----------|-----------|-----------|----------|-----------|-----------|
| F8       | avg    | $-1.25 \times 10^{4}$ | $-1.03 \times 10^{4}$ | $-1.22 \times 10^{4}$ | $-7.60 \times 10^{3}$ | $-9.97 \times 10^{3}$ | $-1.03 \times 10^{4}$ |
| F9       | avg    | 58.7      | $2.05 \times 10^{3}$ | $1.07 \times 10^{3}$ | 8.92 $\times 10^{2}$  | 1.65 $\times 10^{3}$  | 6.42 $\times 10^{2}$  |
| F10      | avg    | $8.88 \times 10^{-16}$ | 4.44 $\times 10^{-15}$ | 9.36 $\times 10^{-04}$ | 2.55      | $8.88 \times 10^{-16}$ | 2.27      |
| F11      | avg    | 0         | 2.25 $\times 10^{-14}$ | 11.9     | 18.8      | 0         | 12.6      |
| F12      | avg    | $3.18 \times 10^{-09}$ | 2.61 $\times 10^{-02}$ | 1.32 $\times 10^{-08}$ | 6.55 $\times 10^{00}$ | 1.81 $\times 10^{-04}$ | 7.60 $\times 10^{-01}$ |
| F13      | avg    | 1.40 $\times 10^{-08}$ | 2.93 $\times 10^{-02}$ | 5.60 $\times 10^{-09}$ | 3.38      | $1.35 \times 10^{-32}$ | 5.15 $\times 10^{-01}$ |

Table 5. Comparison of REWOA with other algorithms for use on fixed-dimension multimodal problems. Note that the bold letters in the table indicate the respective best results.

| Function | Metric | REWOA     | WOA       | WOABAT    | SSA      | WOASAT    | DE        |
|----------|--------|-----------|-----------|-----------|----------|-----------|-----------|
| F14      | avg    | 1.32      | 3.22      | 1.78      | $1.16 \times 10^{-01}$ | 8.14      | 2.51      |
| F14      | std    | 1.75      | 3.47      | 2.50      | $4.50 \times 10^{-01}$ | 5.09      | 2.26      |
| F15      | avg    | $5.37 \times 10^{-04}$ | $8.27 \times 10^{-04}$ | $4.03 \times 10^{-04}$ | $2.96 \times 10^{-03}$ | $5.51 \times 10^{-04}$ | $3.86 \times 10^{-03}$ |
| F15      | std    | $1.67 \times 10^{-04}$ | $5.57 \times 10^{-04}$ | $3.56 \times 10^{-04}$ | $1.11 \times 10^{-02}$ | $3.59 \times 10^{-04}$ | $7.39 \times 10^{-03}$ |
| F16      | avg    | $-1.0316$ | $-1.0316$ | $-1.0316$ | $-1.0316$ | $-1.0316$ | $-1.0316$ |
| F16      | std    | 6.08 $\times 10^{-16}$ | 1.42 $\times 10^{-09}$ | 5.53 $\times 10^{-16}$ | 1.86 $\times 10^{-14}$ | 1.15 $\times 10^{-10}$ | 6.21 $\times 10^{-16}$ |
| F17      | avg    | $3.98 \times 10^{-01}$ | 3.98 $\times 10^{-01}$ | 3.98 $\times 10^{-01}$ | 3.98 $\times 10^{-01}$ | 3.98 $\times 10^{-01}$ | $3.98 \times 10^{-01}$ |
| F17      | std    | 0         | 8.08 $\times 10^{-06}$ | 2.12 $\times 10^{-15}$ | 1.11 $\times 10^{-14}$ | 8.79 $\times 10^{-09}$ | 0         |
| F18      | avg    | 3.90      | 3.00      | 12.0      | $3.00 \times 10^{-0}$  | 3.00      | 5.70      |
| F19      | avg    | 4.85      | 4.82 $\times 10^{-04}$ | 12.7     | 1.79 $\times 10^{-13}$ | 2.21 $\times 10^{-08}$ | 8.10      |
| F19      | avg    | $-3.86$   | $-3.86$   | $-3.86$   | $-3.86$   | $-3.86$   | $-3.86$   |
Table 5. Cont.

| Function | Metric | REWOA | WOA | WOABAT | SSA | WOASAT | DE |
|----------|--------|-------|-----|--------|-----|--------|----|
| F20      | avg    | −3.26 | −3.23 | −3.29 | −3.21 | −3.26 | −3.27 |
|          | std    | 5.94 $\times 10^{-02}$ | 9.53 $\times 10^{-02}$ | 5.06 $\times 10^{-02}$ | 5.70 $\times 10^{-02}$ | 5.93 $\times 10^{-02}$ | 5.93 $\times 10^{-02}$ |
| F21      | avg    | −7.38 | −8.52 | −9.48 | −7.96 | −5.40 | −5.56 |
|          | std    | 3.07  | 2.48 | 1.72  | 2.74  | 1.27  | 3.17 |
| F22      | avg    | −8.55 | −8.05 | −9.17 | −8.38 | −5.09 | −5.29 |
|          | std    | 2.86  | 3.20 | 2.24  | 3.15  | 2.62 $\times 10^{-07}$ | 3.17 |
| F23      | avg    | −8.66 | −7.11 | −10.0 | −8.42 | −5.31 | −6.34 |
|          | std    | 3.19  | 3.51 | 1.61  | 3.30  | 9.71 $\times 10^{-01}$ | 3.52 |

| best sd-best | 4 | 2 | 6 | 4 | 2 | 4 |

4.1. Evaluation of Exploitation Capability

The unimodal functions (F1–F7) can be evaluated using the exploitation capability of the algorithms. The optimizer comparison results and the best and second-best statistical performance are shown in Table 3. Note that the bold letters in the table indicate the respective best results. According to the results in Table 3, in the basic algorithms (WOA, SSA, DE), only in function F6 out of the seven functions is the performance of DE better than REWOA, WOABAT, and WOASAT. The calculation results for WOA and SSA are neither optimal nor suboptimal. WOABAT and WOASAT are optimal in function F7 and functions F1 and F2, respectively. This also verifies the effectiveness of the hybrid optimization strategy. Compared with WOABAT and WOASAT, REWOA has the best optimization performance for functions F3, F4, and F5. It is worth noting that REWOA has the second-best results for functions F1, F2, F6, and F7, which shows its stronger local exploitation power and better stability. To be specific, the random evolutionary strategy enables the optimizer to quickly explore the search space area, and special reinforcement of the development strategy can help the optimal solution to be obtained through further exploitation of the explored space.

4.2. Evaluation of Exploration Capability

Compared with unimodal functions, multimodal functions include many local optima, the number of which increases exponentially with the problem size. Thus, these kinds of problems are often used to evaluate the exploration capability of an algorithm. From Table 4, it can be observed that out of 6 functions, there are 5 functions in which REWOA shows extremely higher competitiveness compared with other algorithms. The mean value and standard deviation calculated by REWOA and WOASAT are both the best in functions F9–F11. However, WOASAT has better results only for F13 and REWOA is better than WOASAT for functions F8 and F12. Therefore, it is concluded that the performance of REWOA is better than WOASAT for multimodal functions. This shows that REWOA can explore the space more adequately and stably, and also has strong ability to escape the local optimum. This is due to the diversity and randomness of the population being increased by the adaptive crossover and improved step-size factor. Furthermore, it can be seen in Table 5 that the fixed-dimension multimodal functions F14–F23 are relatively simple in terms of optimization because of their low dimensionality, so the optimization results are not very different in some functions. In fact, in most benchmark functions, REWOA is always the best or second-best algorithm. These statistical results show that REWOA can conduct a more stable exploration of the unknown solution space and increase the possibility of escaping the local optimum.
4.3. Analysis of Convergence Behavior

In order to analyze the convergence of the proposed algorithm, we selected the convergence curves of 16 representative benchmark problems, as shown in Figure 4. In the graphs, the x-axis indicates the number of iterations and the y-axis represents the best score obtained so far. Because the convergence curves of some functions are similar, we chose representative ones for observation and discussion. For the unimodal functions F1, F2, F5, and F7, it can be observed that REWOA has better convergence compared with WOA, SSA, DE, WOABAT, and WOASAT. For functions F1 and F2, REWOA can rapidly converge to the global optimum at the initial stage, while other algorithms converge more slowly. It can be seen from the function F5 that REWOA falls into the local optimum in the early stage of convergence, similar to WOABAT and WOASAT, while only REWOA escapes the local optimum to find the global optimum in the end. This shows that REWOA exploits better quality solutions to escape from the local optimum. This is because the random evolutionary mechanism allows the optimizer to obtain more information about elite whales to reduce the search range of the population at the initial stages. These convergence results for the unimodal functions show that the proposed algorithm can effectively improve the convergence speed and accuracy. It is very difficult for the optimizer to obtain the global optimum in multimodal problems that have many local optimum solutions. However, REWOA can still maintain the stability of the convergence speed and accuracy. For functions F9 and F10, REWOA can obtain the global optimal solution quickly at the beginning, while SSA and DE converge slowly and obtain a low-precision solution. This is because REWOA uses a weighting factor to divide whale predation strategies into early and late stages. Although the convergence speed of REWOA in functions F8 and F12 is lower than that of WOABAT, WOA, and WOASAT, REWOA can obtain higher quality solutions in the end. This shows that the algorithm has good continuous development capabilities and is always able to explore and exploit the optimal value. This approach should benefit from the hybridization of the evolutionary strategy, which gives the algorithm strong development capabilities. In addition, the convergence curves of most fixed-dimension multimodal functions are basically equivalent. For the fixed-dimension multimodal functions, REWOA’s convergence speed and accuracy are competitive compared to SSA, WOA, and WOASAT. In general, the convergence performance of REWOA further proves that random evolution and special reinforcement can allow the optimizer to find the global optimal solution faster in the search space.

As a summary, the results of this section indicate the performance improvement of the proposed algorithm. The higher exploration capability of REWOA is due to the strengthened whale position updating mechanism using Equation (16). This equation requires whales to move more randomly around each other and increase their distance exponentially. In addition, higher exploitation and convergence are emphasized, which originate from Equations (12), (17), and (21). These equations allow the whales to rapidly reposition themselves under the guidance of the optimal solution. At the same time, the whales utilize Equation (15) to ensure the diversity of the population in order to avoid the local optima. Based on the above analysis, we summarize the limits and highlights of the proposed algorithm. The highlights of the REWOA are that it double-reforms the structure and the location updating mechanism of the basic algorithm. The transformation of the algorithm structure is not just a simple replacement, but rather a redefinition of the operating order and rules. The proposed REWOA has some limits of low convergence accuracy and slow speed in the some fixed-dimension multimodal problems.
Figure 4. Comparison of convergence curves of REWOA and literature algorithms obtained for some of the benchmark problems.
5. Hammerstein Model Identification Using REWOA

5.1. Hammerstein Model

The Hammerstein model [56] is a typical nonlinear system model with a specific structure composed of a nonlinear static block in series with a linear dynamic system block, as shown in Figure 5. The mathematical expression for this model is as follows:

\[ A(z^{-1})y(k) = B(z^{-1})x(k-1) + C(z^{-1})e(k), \]  

where:

\[ A(z^{-1}) = 1 + a_1z^{-1} + \cdots + a_nz^{-n}, \]  
\[ B(z^{-1}) = b_0 + b_1z^{-1} + \cdots + b_rz^{-r}, \]  
\[ C(z^{-1}) = c_0 + c_1z^{-1} + \cdots + c_mz^{-m}, \]

\[ x(k) = F(u(k)) \]  

In this model, \( z^{-1} \) is a unit delay; \( y(k) \) and \( u(k) \) represent the output and input of Hammerstein system samples at instant \( k \), respectively; \( x(k) \) is the output of the nonlinear static part, which is given by Equation (37); \( e(k) \) represents heavy-tailed noise generated by specific distribution function samples at instant \( k \). Function \( F(\cdot) \) represents the static nonlinearity of the Hammerstein model. The symbols \( n, r, \) and \( m \) are the orders of polynomials \( A(z^{-1}), B(z^{-1}), \) and \( C(z^{-1}) \), respectively. The purpose of the identification process is to determine the parameters of the dynamic linear part of the Hammerstein system by using the known input data \( u(k) \) and output data \( y(k) \).

![Figure 5. The Hammerstein model.](image)

In this paper, two kinds of heavy-tailed noise, two-term Gaussian mixture distribution and \( t \)-distribution, are used as the interference signals of the system [57].

(a) Two-term Gaussian mixture distribution

The mathematical expression of the two-term Gaussian mixture distribution is:

\[ e_i(k) \sim (1 - \alpha_i)N_1(\mu_i, \delta_i^2) + \alpha_iN_2(\mu_i, k_i\delta_i^2) \]  

where \( N_1(\mu_i, \delta_i^2) \) represents the Gaussian distribution, for which the mean value and variance are \( \mu_i = 0 \) and \( \delta_i^2 \); \( N_2(\mu_i, k_i\delta_i^2) \) is the impulsive part with the mean value \( \mu_i \) and variance \( k_i\delta_i^2 \). The occurrence probability of the impulse is \( \alpha_i \). It should be noted that \( k_i \gg 1 \).

(b) The \( t \)-distribution

The mathematical expression of the \( t \)-distribution is as follows:

\[ e_i(k) \sim t(\mu_i, \delta_i^2, \nu_i) \]  

\[ e_i(k) \sim (1 - \alpha_i)N_1(\mu_i, \delta_i^2) + \alpha_iN_2(\mu_i, k_i\delta_i^2) \]  

\[ e_i(k) \sim t(\mu_i, \delta_i^2, \nu_i) \]
where \( t(\mu_i, \delta^2_i, v_i) \) represents the univariate Student’s t-distribution; \( \mu_i, \delta^2_i, \) and \( v_i \) represent the location parameter, scale parameter, and degrees of freedom, respectively.

5.2. The Identification Procedure

In this section, the identification problem in the Hammerstein model is converted into a single objective optimization problem. The positions of the whales in REWOA correspond to the possible value of the parameter vector. The fitness value is the mean squared error (MSE), defined as Equation (40). The purpose of the identification task is to minimize the value of the MSE using REWOA.

\[
MSE = \frac{1}{J} \sum_{k=1}^{J} [y(k) - \hat{y}(k)]^2
\]

(40)

where \( J \) is the number of sampled data points; \( y \) is the real output of the system and \( \hat{y} \) is the estimated output of the Hammerstein model. A schematic diagram of the identification procedure in the Hammerstein model using REWOA is shown in Figure 6. The nonlinear static part is modeled with the functional link artificial neural network (FLANN). The structure of the FLANN is shown in Figure 7. The steps of the identification procedure using REWOA as follows:

**Step1:** Obtain the input sample data \( u(k) \) and output sample data \( y(k) \) of the system;

**Step2:** Calculate the output \( \hat{y}(k) \) of the model according to the weight vector of the auxiliary model;

**Step3:** Initialization of positions and parameters;

**Step4:** Minimize the fitness value using REWOA to get the best solution in the current iteration;

**Step5:** Check whether the identification result is satisfied or not. If satisfied, then stop the algorithm and get the best solutions; if not, go back to step 4 and set \( k = k + 1 \).

Figure 6. Schematic diagram of the identification process.
5.3. Simulation Study

In order to verify the feasibility of the above method, three different experiments were conducted to evaluate the performance of the REWOA for parameter identification. The identification results were compared with WOA, WOABAT, DE, and PSO. To make a fair comparison, the maximum iteration number for all algorithms was set to 300, the population size was set to 30, and the control parameter values of all the algorithms were derived from the literature. The number of nonlinear system data samples collected was 50. All experiments were conducted in MATLAB R2019a environment on a 64 bit Windows 10 PC with an Intel i5 9300H processor and 16 GB RAM.

- Experiment 1

Consider the following Hammerstein model:

\[ A(z^{-1})y(k) = B(z^{-1})x(k - 1) + C(z^{-1})e(k), \]  
\[ A(z) = 1 + 0.8z^{-1} + 0.6z^{-2}, \]  
\[ B(z) = 0.4 + 0.2z^{-1}, \]  
\[ C(z) = 1 \]  
\[ X(k) = u(k) + 0.5u^3(k) \]  

The input signal \( u(k) \) is uniformly distributed in the range of \([-3, 3]\). The interference \( e(k) \) is a Gaussian noise with zero mean and a variance of 0.01. The nonlinear part is modeled by the FLANN structure and each input signal is expanded as \( u(n) = [1, u(n), u^2(n), u^3(n)]^T \) by a power series, while the output of the model is given as:

\[ \phi[u(k)] = u^T(n)w_1(k) \]  

Figure 8 shows the convergence curves for REWOA, WOA, WOABAT, DE, and PSO. This figure shows that convergence of the REWOA is better than other algorithms and the output of model for REWOA can fit the actual output of the system well.
Figure 8. The MSE convergence curve for experiment 1.

- **Experiment 2**
  
  Consider the following Hammerstein model:
  
  $$A(z^{-1})y(k) = B(z^{-1})x(k-1) + C(z^{-1})e(k),$$  
  
  $$A(z) = 1 + 0.9z^{-1} + 0.15z^{-2} + 0.02z^{-3},$$  
  
  $$B(z) = 0.7 + 1.5z^{-1}$$  
  
  $$C(z) = 1$$  
  
  $$X(k) = 0.5 \sin[\pi u(k)]$$  
  
  $$e(k) \sim (1 - \alpha)N_1(u, \sigma^2) + \alpha N_2(u, k\sigma^2)$$  

  The input signal $u(k)$ is uniformly distributed in the range of $[-3, 3]$. The interference $e(k)$ is a two-term Gaussian mixture distribution heavy-tailed noise, for which $\alpha = 0.05$, $\sigma^2 = 0.2^2$, and $k = 100$. The nonlinear part is modeled by the FLANN structure and each input signal is expanded as:
  
  $$u(n) = \{1, u(n), \sin[\pi u(n)], \cos[2\pi u(n)], \sin[3\pi u(n)]\}^T$$  

  The output of the model is given as:
  
  $$\phi[u(n)] = \tanh[u^T(n)w_1(n)]$$  

  Figure 9 shows the MSE convergence curves for REWOA, WOA, WOABAT, DE, and PSO. It should be noted that the identification results for REWOA are more accurate than for WOABAT and WOA. Under the influence of heavy-tail noise, the convergence ability of REWOA is much better than the other four comparison algorithms; it is possible that these algorithms fall into the trap of obtaining local optimal solutions.
Experiment 3

Consider the following Hammerstein model:

\[ A(z^{-1})y(k) = B(z^{-1})x(k - 1) + C(z^{-1})e(k), \]
\[ A(z) = 1 - 1.5z^{-1} + 0.7z^{-2} \]
\[ B(z) = z^{-1} + 0.5z^{-2} \]
\[ C(z) = 1 - 0.5z^{-1} \]
\[ X(n) = u(n) + 0.5u^2(n) + 0.3u^3(n) + 0.1u^4(n) \]
\[ e(k) \sim t(0, 0.2^2, 6) \]

The input signal \( u(k) \) is uniformly distributed in the range of \([-1, 1]\), while the interference signal \( e(k) \) is the heavy-tailed noise of the t-distribution. The nonlinear part is modeled by the FLANN structure and each input signal is expanded as Equation (61).

\[ u(n) = \{1, u(n), \sin[u(n)], \cos[2u(n)], \sin[3u(n)]\}^T \]

The output of the model is given by Equation (62).

\[ \phi[u(n)] = u^T(n)w_1(n) \]

In experiment 3, due to the influence of the heavy-tailed noise, the accuracy of the convergence for REWOA decreased, as shown Figure 10. Even so, Figure 10 shows that the convergence accuracy for REWOA is the highest. REWOA can still meet the requirements for identification and performs the best, which also shows the feasibility of our proposed method.
Figure 11. The residual test results for the three experiments: (a) experiment 1; (b) experiment 2; (c) experiment 3.

According to the residual results, it was observed that the method of identification for the Hammerstein model using intelligent optimization algorithm is reliable and feasible, and it is able to reflect the characteristics of the actual system. In addition, this method solves the problem of there being no uniform modus for nonlinear system identification. Therefore, it is important and worthy of further research.
6. Conclusions

In this paper, we proposed an improved WOA based on the hybridization of random evolutionary strategies and special strengthening strategies in different dimensions. The newly proposed REWOA enhances the performance of the algorithm through the improvement and combination of the structure and the location updating mechanism. Firstly, the three location updating mechanisms of the WOA algorithm were strengthened. Secondly, a different random evolution strategy was added to the structure of the enhanced WOA algorithm in different dimensions. These changes improved the balance ability for exploration and exploitation and overcame the low convergence accuracy and slow convergence speed of the pure WOA. Based on the testing of 23 benchmark functions, the calculation performance of the REWOA was verified by comparing it with other well-known metaheuristics and WOA variants. The obtained results indicate that REWOA can find higher quality solutions and converge faster in most benchmark functions, improving the stability and reliability of the algorithm and further enhancing its computing power. In addition, the proposed REWOA was applied to a general scheme of a Hammerstein model identification. The nonlinear part of the Hammerstein process was modeled using a functional link artificial neural network and the identification problem was transformed into an optimization problem. Three experiments were performed involving parameter identification for the Hammerstein model under two different heavy-tailed noise conditions and the results were compared with other techniques, verifying the competitiveness of the proposed REWOA in terms of its performance. The obtained residual test results further verify the feasibility, effectiveness, and superiority of using REWOA to solve the parameter identification problem in nonlinear systems.

Author Contributions: Conceptualization, Q.J. and Z.X.; methodology, Q.J.; software, Z.X.; validation, Q.J., Z.X. and W.C.; formal analysis, Z.X.; investigation, Z.X.; resources, Q.J.; data curation, Q.J. and Z.X.; writing—original draft preparation, Z.X.; writing—review and editing, Q.J., Z.X. and W.C.; visualization, Q.J.; supervision, Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study did not involve humans or animals.

Informed Consent Statement: The study did not involve humans or animals.

Data Availability Statement: In the paper, all the data generation information has been given in detail in the related chapter.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Greiner, R. PALO: A probabilistic hill-climbing algorithm. Artif. Intell. 1996, 84, 177–208. [CrossRef]
2. Solis, F.J.; Wets, R.J.B. Minimization by Random Search Techniques. Math. Oper. Res. 1981, 6, 19–30. [CrossRef]
3. Ouyang, W.; Zhang, B. Newton’s method for fully parameterized generalized equations. Optimization 2018, 67, 2061–2080. [CrossRef]
4. Jamil, M. Levy Flights and Global Optimization. In Swarm Intelligence and Bio-Inspired Computation: Theory and Applications; Elsevier: Amsterdam, The Netherlands, 2013. [CrossRef]
5. Gandomi, A.H.; Yang, X.S.; Talatahari, S.; Alavi, A.H. Metaheuristic Algorithms in Modeling and Optimization. Metaheuristic Appl. Struct. Infrastruct. 2013, 1–24. [CrossRef]
6. Yang, X.-S. Nature-Inspired Metaheuristic Algorithms; Luniver Press, Middlesex University: London, UK, 2010.
7. Yang, X.S. Firefly Algorithm, Levy Flights and Global Optimization. In Research and Development in Intelligent Systems XXVI; Springer: London, UK, 2010.
8. Holland, J.H. Genetic Algorithms. Sci. Am. 1992, 267, 66–72. [CrossRef]
9. Simon, D. Biogeography-Based Optimization. IEEE Trans. Evol. Comput. 2009, 12, 702–713. [CrossRef]
10. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. IEEE Trans. Evol. Comput. 2011, 15, 4–31. [CrossRef]
11. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
12. Dorigo, M.; Birattari, M.; Stützle, T. Ant Colony Optimization. *IEEE Comput. Intell. Mag.* 2006, 1, 28–39. [CrossRef]
13. Karaboga, D.; Bastürk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* 2007, 8, 687–697. [CrossRef]
14. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* 2012, 17, 4831–4845. [CrossRef]
15. Feng, Y.; Wang, G.G.; Deb, S.; Lu, M.; Zhao, X.J. Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Comput. Appl.* 2017, 28, 1619–1634. [CrossRef]
16. Wang, G.; Deb, S.; Coelho, L. Earthworm optimization algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Intern. J. Bio-Inspir. Comput.* 2018. [CrossRef]
17. Wang, G.G.; Deb, S.; Coelho, L.D.S. Elephant Herding Optimization. In Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBII 2015), Bali, Indonesia, 7–9 December 2015; IEEE: Piscataway, NJ, USA, 2016.
18. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* 2019, 97, 849–872. [CrossRef]
19. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* 2020, 111, 300–323. [CrossRef]
20. Wang, G.G. Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Futur. Gener. Comput. Syst.* 2020, 111, 300–323. [CrossRef]
21. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* 2016, 95, 51–67. [CrossRef]
22. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* 2017, 114, 163–191. [CrossRef]
23. Wu, G.; Mallipeddi, R.; Suganthan, P.N.; Wang, R.; Chen, H. Differential evolution with multi-population based ensemble of mutation strategies. *Inf. Sci. Int. J.* 2016, 329, 329–345. [CrossRef]
24. Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Trans. Evol. Comput.* 2009, 13, 398–417. [CrossRef]
25. Olorunda, O.; Engelbrecht, A.P. Measuring Exploration/Exploitation in Particle Swarms using Swarm Diversity. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 1128–1134.
26. Alba, E.; Dorronsoro, B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evol. Comput.* 2005, 9, 126–142. [CrossRef]
27. Lin, L.; Gen, M. Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation. *Soft Comput.* 2009, 13, 157–168. [CrossRef]
28. Heidari, A.A.; Aljarah, I.; Faris, H.; Chen, H.; Luo, J.; Mirjalili, S. An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput. Appl.* 2019, 32, 5185–5211. [CrossRef]
29. Liu, S.H.; Mernik, M.; Hrnčič, D.; Crepinšek, M. A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova’s mass transfer model. *Appl. Soft Comput.* 2013, 13, 3792–3805. [CrossRef]
30. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* 2014, 69, 46–61. [CrossRef]
31. Vesselinov, V.V.; Harp, D.R. Adaptive hybrid optimization strategy for calibration and parameter estimation of physical process models. *Comput. Geosci.* 2012, 49, 10–20. [CrossRef]
32. Li, L.; Wang, L.; Liu, L.-H. An effective hybrid PSOSA strategy for optimization and its application to parameter estimation. *Appl. Math. Comput.* 2006, 179, 135–146. [CrossRef]
33. Dzemyda, G.; Senkienė, E.; Valeyvicienė, J. Simulated annealing for parameter grouping. *Informatica* 1990, 1, 20–39.
34. Mo, W.; Guan, S.G.; Puthusserypady, S.K. A Novel Hybrid Algorithm for Function Optimization: Particle Swarm Assisted Incremental Evolution Strategy. *Stud. Comput. Intell.* 2007, 75, 101–125. [CrossRef]
35. Ghanem, W.A.; Jantan, A.B. Hybridizing artificial bee colony with monarch butterfly optimization for numerical optimization problems. *Neural Comput. Appl.* 2018, 30, 163–181. [CrossRef]
36. Javaid, N.; Ullah, I.; Zarin, S.S.; Kamal, M.; Omoniwa, B.; Mateen, A. Differential-evolution-earthworm hybrid meta-heuristic optimization technique for home energy management system in smart grid. In Proceedings of the 12-th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018), Matsue, Japan, 4–6 July 2018; Springer: Piscataway, NJ, USA, 2016.
37. Prakash, D.; Lakshminarayana, C. Optimal siting of capacitors in radial distribution network using Whale Optimization Algorithm. *Alex. Eng. J.* 2017, 56, 499–509. [CrossRef]
38. Kumar, C.S.; Rao, R.S.; Cherukuri, S.K.; Rayapudi, S.R. A Novel Global MPP Tracking of Photovoltaic System based on Whale Optimization Algorithm. *Int. J. Renew. Energy Dev.* 2016, 5, 225. [CrossRef]
39. Miao, Y.; Zhao, M.; Makis, V.; Lin, J. Optimal swarm decomposition with whale optimization algorithm for weak feature extraction from multicomponent modulation signal. *Mech. Syst. Signal Process.* 2019, 122, 673–691. [CrossRef]
40. Petrović, M.; Miljković, Z.; Jokić, A. A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm. *Appl. Soft Comput.* 2019, 81, 105520. [CrossRef]
41. Aljarah, I.; Faris, H.; Mirjalili, S. Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Comput. 2018, 22, 1–15. [CrossRef]
42. Bhatt, U.R.; Dhakad, A.; Chouhan, N.; Upadhay, R. Fiber wireless (FiWi) access network: ONU placement and reduction in average communication distance using whale optimization algorithm. Heliyon 2019, 5, e01311. [CrossRef]
43. Mohammed, H.M.; Umar, S.U.; Rashid, T.A. A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm. Comput. Intell. Neuroscience 2019. [CrossRef]
44. Bozorgi, S.M.; Yazdani, S. IWOA: An improved whale optimization algorithm for optimization problems. J. Comput. Des. Eng. 2019, 6, 243–259.
45. Singh, A. Laplacian whale optimization algorithm. Int. J. Syst. Assur. Eng. Manag. 2019, 10, 713–730. [CrossRef]
46. Sayed, G.I.; Darwish, A.; Hassanien, A.E. A New Chaotic Whale Optimization Algorithm for Features Selection. J. Classif. 2018, 35, 300–344. [CrossRef]
47. Fan, Q.; Chen, Z.; Zhang, W.; Fang, X. ESSAWOA: Enhanced Whale Optimization Algorithm integrated with Salp Swarm Algorithm for global optimization. Eng. Comput. 2020, 1–18. [CrossRef]
48. Yan, Z.; Zhang, J.; Tang, J. Modified whale optimization algorithm for underwater image matching in a UUV vision system. Multimed. Tools Appl. 2021, 80, 187–213. [CrossRef]
49. Salgotra, R.; Singh, U.; Saha, S. On Some Improved Versions of Whale Optimization Algorithm. Arab. J. Sci. Eng. 2019, 44, 9653–9691. [CrossRef]
50. Elghamrawy, S.M.; Hassanien, A.E. GWOA: A hybrid genetic whale optimization algorithm for combating attacks in cognitive radio network. J. Ambient. Intell. Hum. Comput. 2018, 10, 4345–4360. [CrossRef]
51. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. Neurocomputing 2017, 260, 302–312. [CrossRef]
52. Barthelemy, P.; Bertolotti, J.; Wiersma, D.S. A Levy flight for light. Nature 2008, 453, 495–498. [CrossRef]
53. Yanming, D.; Huihui, X.; Fang, L. Flower pollination algorithm with new pollination methods. Comput. Eng. Appl. 2018, 54, 94–108.
54. Papadimitriou, F. The Probabilistic Basis of Spatial Complexity. In Spatial Complexity, Theory, Mathematical Methods and Applications; Springer: Cham, Switzerland; Metro Manila, Philippines, 2020.
55. Gutowski, M. Levy flights as an underlying mechanism for global optimization algorithms. 2001, arXiv:math-ph/0106003. Available online: https://arxiv.org/abs/math-ph/0106003 (accessed on 30 January 2021).
56. Kung, F-C; Shih, D.-H. Analysis and identification of Hammerstein model non-linear delay systems using block-pulse function expansions. Int. J. Control 2007, 43, 139–147. [CrossRef]
57. Kumar, T.A.; Deergha Rao, K. A new m-estimator based robust multiuser detection in flat-fading non-gaussian channels. IEEE Trans. Commun. 2009, 57, 1908–1913. [CrossRef]