Hindawi

Research Article

K-Means Segmentation of Underwater Image Based on Improved Manta Ray Algorithm

Donglin Zhu [iD],[1,2] Linpeng Xie,[1] and Changjun Zhou [iD][1]

[1]*College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China*
[2]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China*

Correspondence should be addressed to Changjun Zhou; zhouchangjun@zjnu.edu.cn

Image segmentation plays an important role in daily life. The traditional K-means image segmentation has the shortcomings of randomness and is easy to fall into local optimum, which greatly reduces the quality of segmentation. To improve these phenomena, a K-means image segmentation method based on improved manta ray foraging optimization (IMRFO) is proposed. IMRFO uses Lévy flight to improve the flexibility of individual manta rays and then puts forward a random walk learning that prevents the algorithm from falling into the local optimal state. Finally, the learning idea of particle swarm optimization is introduced to enhance the convergence accuracy of the algorithm, which effectively improves the global and local optimization ability of the algorithm simultaneously. With the probability that K-means will fall into local optimum reducing, the optimized K-means hold stronger stability. In the 12 standard test functions, 7 basic algorithms and 4 variant algorithms are compared with IMRFO. The results of the optimization index and statistical test show that IMRFO has better optimization ability. Eight underwater images were selected for the experiment and compared with 11 algorithms. The results show that PSNR, SSIM, and FSIM of IMRFO in each image are better. Meanwhile, the optimized K-means image segmentation performance is better.

1. Introduction

In recent years, image segmentation has attracted much attention and research by researchers. It is of great significance to the future image processing field. As a key step of image processing, image segmentation plays an important role in extracting objects of interest from images. At present, it has important research value in medicine, agriculture, ocean, and other fields. Image segmentation can be divided into four categories: threshold segmentation, region segmentation, edge segmentation, and segmentation methods based on specific theories. The clustering algorithm is a typical unsupervised learning algorithm. It uses the idea of clustering differentiation to solve the problem. The way to solve the problem is simple and easy to understand. It has been successfully applied in many fields [1]. Cluster image segmentation has also been successfully studied. K-means is the most common and easiest clustering method among them, but K-means has the disadvantage of large randomness and easily fall into local optimum, which makes it impossible to control the cluster center reasonably. Swarm intelligence algorithms is an algorithm with global optimization performance and strong versatility that is suitable for parallel processing. This type of algorithm can find the optimal solution or approximate the optimal solution within a certain period of time. [2] Intelligent optimization algorithm opens up a new way for image segmentation. In terms of clustering segmentation, Hrosik R C and others improved the K-means clustering algorithm based on the firefly algorithm, which could achieve better segmentation average error, peak signal-to-noise ratio, and structural similarity index on medical images; [3] Li h and others proposed a k-means clustering algorithm based on dynamic particle swarm optimization (DPSO), which had better visual effect than traditional K-means clustering in image segmentation and obvious advantages in improving image segmentation quality and efficiency; [4] Shubham and others applied gray wolf optimizer (GWO) [5] to the segmentation of satellite images; [6] Therefore, an intelligent optimization algorithm is of great significance in the field of image segmentation.

With the rapid development of swarm intelligence algorithms, a variety of new algorithms are emerging. In addition to the algorithms mentioned above, there are other algorithms as follows: monarch butterfly optimization (MBO) [7], elephant herding optimization (EHO) [8], moth search (MS) algorithm [9], Harris hawks optimization (HHO) [10], etc. Manta ray foraging optimization (MRFO) is a new swarm intelligence optimization algorithm proposed in 2020. With excellent searchability, fewer parameters, simple model and easily understood, it is better than particle swarm optimization (PSO) [11, 12], genetic algorithm (GA) [13, 14], Differential Evolution (DE) [15, 16], Cuckoo Search (CS) [17], gravitational search algorithm (GSA) [18], and ABC [19] in some function optimization [20], and it has been successfully applied to solar energy [21, 22], ECG [23], generator [24, 25], power system [26], cogeneration energy system [27], geophysical inversion problem [28], directional overcurrent relay [29], feature selection [30], hybrid energy system [31], and sewage treatment [32]. Although MRFO has good optimization ability, it still has its own defects. In complex problems, the search ability is limited and the diversity of the population can not be guaranteed. The main reason is that the individual searches orderly, highly dependently, and inflexibly.

At present, researchers have noticed this point and carried out successive studies, such as Mohamed Abd Elaziz, who combines fractional calculus with MRFO to correct the direction of manta ray movement. This algorithm has been verified by CEC 2017 test function and is applied to image segmentation problems with good feasibility [33]. Mohamed H. Hassan combines a gradient optimizer with MRFO to reduce the probability that the algorithm will fall into a local optimum, which has a good effect in single-objective and multiobjective economic emission scheduling [34]. Haitao Xu uses adaptive weighting and chaos to improve MRFO, so as to efficiently handle thermodynamic problems [35]. Essam H. Houssein uses reverse learning to initialize the population so as to increase the diversity of the population and apply it to the threshold image segmentation problem with good segmentation quality [36]. Bibekananda Jena adds an attack capability to MRFO, which allows it to jump out of local optimization and find a globally optimal solution. It is then applied to the image segmentation problem of 3D Tsallis [37]. Mihailo Micev fuses Simulate Anneal (SA) with MRFO and applies it to the Proportional Integral Derivative (PID) controller. The fused algorithm is superior to other algorithms [38]. In addition, Serdar and others adopt opposition-based learning and SA to improve the convergence effect of MRFO. It has better control performance when applied to fractional order proportional integral derivative (FOPID) controller [39]. Although the currently proposed variants of MRFO have achieved some results, the following problems still exist:

(1) Most scholars use the fusion of other algorithms to improve the search ability, but this will bring higher time complexity, and the algorithm after fusion may not be able to complement each other so as to present perfect results.

(2) Reverse learning can only solve inversely in a certain space, but in complex high-dimensional situations, there are fewer individual optimization methods, and they cannot jump out of the local optimal state perfectly.

(3) In the optimization process, the above algorithm cannot completely balance the local search and global search capabilities, which results in insufficient convergence accuracy of the algorithm.

Based on the above analysis, this paper presents an improved algorithm for manta rays, which uses random walk learning to make individuals wander randomly in space, to increase the diversity of the population, and avoid premature convergence of the algorithm, and then we use Lévy flight for long-distance and short-distance searches to balance the local and global searches of the algorithm. Finally, the learning idea of particle swarm is introduced. Two learning factors are used to improve the convergence accuracy of the algorithm 12 functions are used to verify the validity and feasibility of IMRFO. Then eight underwater image datasets are used in K-means image segmentation. The results show that IMRFO has better generalization ability and better segmentation quality.

The innovations and contributions of this paper are as follows:

(i) A random walk learning algorithm is designed to increase the diversity of the population and reduce the probability of the algorithm falling into local optimization to a certain extent.

(ii) Lévy flight and learning factors are introduced to balance the searchability of the algorithm, which makes the algorithm have a good convergence effect.

(iii) In 12 standard test functions, IMRFO is compared with 7 other algorithms to show its superiority and feasibility. Next, two statistical tests are used to emphasize the optimization performance of the algorithm. It is compared with the recently proposed variants of the algorithm. Finally, ablation experiments were performed, all the results show that IMRFO has a good search ability.

(iv) IMRFO is applied to K-means underwater image segmentation. The results of 11 algorithms show that IMRFO performs well.

The structure of this paper is as follows: Section 2 introduces the basic MRFO algorithm, Section 3 introduces the improved IMRFO algorithm and related analysis. Section 4 describes the process of IMRFO optimizing K-means image segmentation. Section 5 tests the performance of IMRFO and compares the related algorithms. Section 6 describes and analyses the performance of each algorithm in K-means image segmentation. Section 7 summarizes the experimental results of this paper. The last section expresses the advantages and disadvantages of IMRFO and future research directions.

2. Manta Ray Foraging Optimization

Manta rays feed on plankton, which are mainly water microfauna. When feeding, they suck water and prey into their mouths with angular head leaves. They then filter prey out of the water through improved rabbles. Individuals of the manta rays work together to find the best food. Inspired by the behavior of the manta rays, the algorithm is divided into chain feeding, spiral-feeding, and somersault foraging. There are three stages of spiral and empty foraging.

2.1. Chain Feeding. At this stage, the manta ray population will be arranged in an ordered chain to collaborate in feeding, which will maximize the amount of plankton in the pocket. The mathematical model of the chain feeding process can be expressed as follows:

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{1}$$

In formula (1), $x_i^d(t)$ denotes the d-dimensional information of the location of the first manta ray in generation t. R is a random number that obeys a uniform distribution of [0,1]. $\alpha = 2 \cdot r \cdot \sqrt{|\log(r)|}$ is the weight factor, $x_{\text{best}}^d(t)$ is the d-dimensional information of the best location found so far The manta ray at position i depends on the manta ray at position i-1 and the best food position found so far. N represents the population number. The update of the first manta ray depends on the optimal location.

2.2. Spiral Feeding. When a manta ray finds a good food source in a certain space, each individual approaches a manta ray in front of it, in addition to spirally moving toward the food. The spiral-feeding process can be represented by the following mathematical model:

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t) + r \cdot \left(x_{best}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right), & i = 1, \\ x_{best}^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{2}$$

where $\beta = 2e^{r_1(T-t+1)/T} \cdot \sin(2\pi r_1)$, a weight factor representing the spiral motion, T being the largest number of iterations, r being the rotation factor and obeying [0,1] uniform random numbers. In addition, in order to improve the efficiency of population foraging, MRFO randomly generates a new location during the optimization process and then performs a spiral search at that location. Its mathematical model is as follows:

$$x_i^d(t+1) = \begin{cases} x_{\text{rand}}^d(t) + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_{\text{rand}}^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{3}$$

$x_{\text{rand}}^d(t)$ represents a new location in space.

2.3. Somersault Foraging. When a certain manta ray finds a food source, its position can be regarded as an axis. Each manta ray tends to wander around the axis and flip to a new location. Its mathematical model is as follows:

$$x_i^d(t+1) = x_i^d(t) + S \cdot \left(r_2 \cdot x_{\text{best}}^d(t) - r_3 \cdot x_i^d(t)\right), i = 1, 2, \ldots, N. \tag{4}$$

S is the flip factor, which determines the flip distance. R_2 and r_3 are two random numbers that are uniformly distributed [0,1]. As S values vary, individual mantas flip to locations in search space that are symmetrical to the optimal solution at their current location.

3. Improved Manta Ray Foraging Optimization

From the above formulas, it can be seen that more communication between individuals and orderly work can improve the searchability of the algorithm and perform a wide-ranging search. On the one hand, the lack of initiative of individuals in the population limits their ability to develop. On the other hand, updates within the population are related to the best location. When encountering high-dimensional complex problems, the change of the optimal position is similar, which results in less change in the two updates before and after the algorithm, which limits the algorithm's optimization ability. Therefore, a flexible change strategy is needed to improve the development ability and local convergence effect of the algorithm. This paper uses the

Lévy flight strategy to improve individual blindness search, and random walk learning is used to prevent the algorithm from falling into a local state and the learning idea of particle swarm to improve the search accuracy of the algorithm.

3.1. Why Each Modification Has Been Proposed?

MRFO is based on a group of animals collaborating in feeding, which results in fewer optimization methods and a lack of flexibility and fineness. Therefore, individual initiative is required to increase the diversity of the population in order to find high-quality solutions in space.

Therefore, this paper analyses and solves the defects of the algorithm from the following three points. Firstly, it is necessary to make the population individuals better distribute the whole space so as to develop the vision of the algorithm and improve the global search ability of the algorithm. Lévy flight is a classical strategy, which can fly in a given space in the way of alternating long and short distances. It has been used by most scholars to improve the search ability of the algorithm. Secondly, some individuals need to be independent and never be limited by group characteristics. Random walk learning is an uncertain way of walking. The traditional random walk can only be carried out in local areas. However, the random walk learning designed in this paper can make large location differences between different individuals and improve the population diversity of the algorithm. Finally, information sharing among individuals is needed to improve the local search ability of the algorithm and find high-quality solutions. The learning factor is derived from the particle swarm optimization algorithm, which is used to speed up the information exchange of the population, prevent the early invalid search, improve the local search ability of the algorithm, and improve the accuracy of the solution to a certain extent.

3.2. Lévy Flight Strategy.

When manta ray individuals perform chain search, all individuals follow the population to search, which leads to the lack of flexibility of the algorithm and can not perform a better search range. Therefore, the Lévy flight strategy [40, 41] is introduced to enable individuals to search long and short distances, increase the diversity of the population, and enable individuals to fully diffuse into the whole space. The location update format for joining Lévy flight strategy is as follows:

$$x_i'(t) = x_i(t) + l \oplus \text{levy}(\lambda). \tag{5}$$

In formula (8), $x_i(t)$ represents the position of the i-th individual in the t-th iteration, $\oplus$ is an arithmetic symbol representing point-to-point multiplication. $l = 0.01(x_i(t)-x_p)$ denotes a step length control parameter, x_p represents the position of the best individual in the population.

Lévy flight formula is as follows [42]:

$$\text{Levy}(x) = 0.01 \times \frac{r_4 \times \sigma}{|r_5|^{(1/\xi)}}. \tag{6}$$

where r_4 and r_5 are random numbers within the range of [0,1], ξ. The general value is 1.5. σ is calculated as follows:

$$\sigma = \left(\frac{\Gamma(1+\xi) \times \sin(\pi\xi/2)}{\Gamma(1+\xi/2) \times \xi \times 2^{(\xi-1/2)}} \right)^{(1/\xi)}. \tag{7}$$

where $\Gamma(x) = (x-1)!$, the schematic diagram of Lévy flight is shown in Figure 1. Lévy flight can search long and short distances in a certain space and balance the global and local search of the algorithm.

3.3. Random Walk Learning.

In the optimization process, MRFO has the probability of falling into the local optimum, which makes the current optimum individual unreliable, so it is necessary to disperse all the individuals to find a better solution. Unlike random walks, the learning factors at the best and worst locations are introduced to make individual escape directional and reduce unreasonable walk. The specific mathematical model of RWL is as follows:

$$x_{i,j}^{t+1} = x_{\text{best}}^t + \left(\frac{2}{1 + t/M \cdot \sin(\pi/2 \cdot r)} - 1 \right)$$
$$\cdot e^{\left(-\frac{t}{M} \right) \cdot \left(x_{i,j}^t - x_{\text{worst}}^t \right)} + \left(c_1 \cdot x_{\text{best}}^t - c_2 \cdot X_{i,j}^t \right). \tag{8}$$

In formula (8), $((2/1 + t/M. \sin(\pi/2.r)) - 1)$ is the sinusoidal random factor that uses the mathematical properties of the sinusoidal function to fluctuate toward the optimal solution and continuously adjusts the step size based on the worst position of the current population so that the search path can span the entire solution space. M is the maximum number of iterations, and c_1 and c_2 represent two learning factors, random numbers that obey a normal distribution. $e^{(-t/M) \cdot (x_{i,j}^t - x_{\text{worst}}^t)}$ is the control step, $(c_1 . x_{\text{best}}^t - c_2 . X_{i,j}^t)$ is the direction of control. As shown in Figure 2(a), (a) is the distribution of individuals without introducing RWL, (b) introduces the individual distribution of RWL; we can see that the introduction of RWL enables individuals to master global information, makes the individual distribution more even, and finds the global optimal solution.

3.4. PSO Algorithm Learning Ideas.

There are two learning factors in PSO to develop local solutions, which can effectively improve the convergence accuracy of the algorithm. Therefore, the formula of introducing two learning factors is as follows:

$$x_{i,j}^{t+1} = b_1 \cdot x_{i,j}^t + b_2 \cdot \text{rand}(1) \cdot \left(\text{BestX} - x_{i,j}^t \right). \tag{9}$$

b_1, b_2 are two learning factors, and *BestX* is the optimal position of the current population. As can be seen from the formula, this strategy exploits individuals between the current one and the optimal one to enhance the local search of the algorithm.

3.5. Improved Manta Ray Foraging Optimization.

To improve the local search capability of MRFO and reduce the probability of falling into local optimum, an improved manta ray algorithm is presented in this paper. The
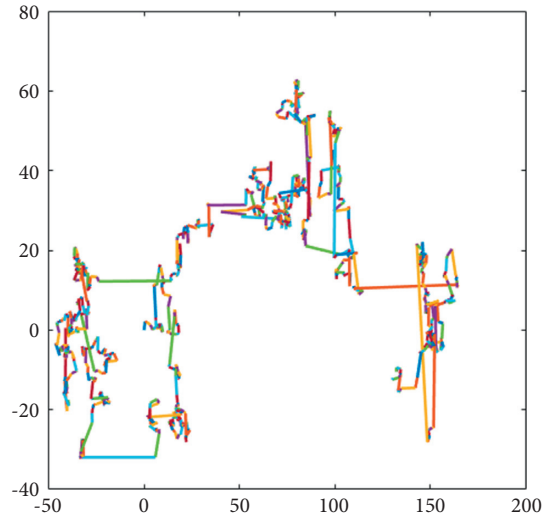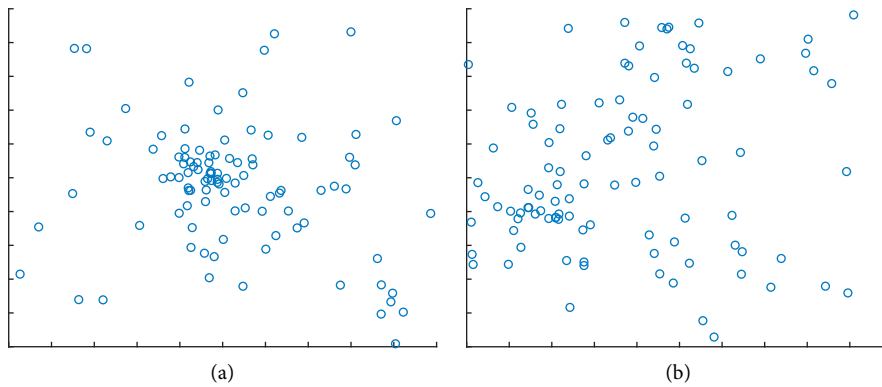
FIGURE 1: Lévy flight diagram.

FIGURE 2: Distribution of algorithm individuals (a) original algorithm (b) algorithm with GWL.

algorithm uses random walk learning to prevent the algorithm from falling into the local state after each iteration and to improve the global search ability of the algorithm. Then, the Lévy flight mechanism is combined to improve the blindness of the manta ray algorithm and to balance the searchability of the algorithm. Using two learning factors of particle swarm optimization to improve the search accuracy ultimately makes the algorithm improve effectively both in local and global aspects. The specific pseudocode is shown in Algorithm 1.

4. K-Means Image Segmentation Based on IMRFO

The principle of the traditional K-means algorithm is to select K cluster centers randomly, so the way to select them is uncertain, resulting in large differences in the final results and easy to fall into local optimum. Therefore, it is necessary to select an appropriate initial cluster center. Intelligent optimization algorithm has been successfully applied to K-means to improve its randomness and the defect of falling into local optimum. The improved manta ray foraging optimization optimizes K-means so that the initial cluster

centers are well controlled. The objective function is as follows:

$$f = \sum_{i=1}^{n} \sum_{j=1}^{m} |X_i - Y_i|^2. \tag{10}$$

X_i is a pixel gray value of the image and Y_i is the J-th clustering center. The optimal initial number of clustering centers is obtained by IMRFO to minimize the fitness value of the objective function.

K-means image segmentation based on IMRFO is mainly divided into two parts:

(1) Use the global search capabilities of IMRFO to find the best initial cluster center in the image point set

(2) The initial cluster centers of the IMRFO output are segmented in the K-means algorithm

The specific flow chart is shown in Figure 3.

5. Performance Analysis and Test

5.1. Performance Test. To verify the effectiveness and feasibility of IMRFO, 12 benchmark functions [43, 44] are

 Input
M : Maximum number of iterations
N : Population sparrows
Output : X_{best}, f_g
Initialize population
$t = 1$;
While($t < M$)
Update the position of the population according to formula (8)%RWL
For $i = 1$ to N do
If rand<0.5 then
If $t/M <$ rand then
Update the position of the individual according to formula (3)
Else
Update the position of the individual according to formula (2)
End if
Else
Update the position of the individual according to formula (1)
End if
Update the position of the population according to formula (5)% Lévy flight
Get the current best and worst
For $i = 1$ to N do
Update the position of the individual according to formula (4)
End for
The position of the individual is updated according to formula (9)% PSO algorithm learning ideas
Get a new Xbest, fg, X_{worst};
End while
Get the final Xbest, fg.
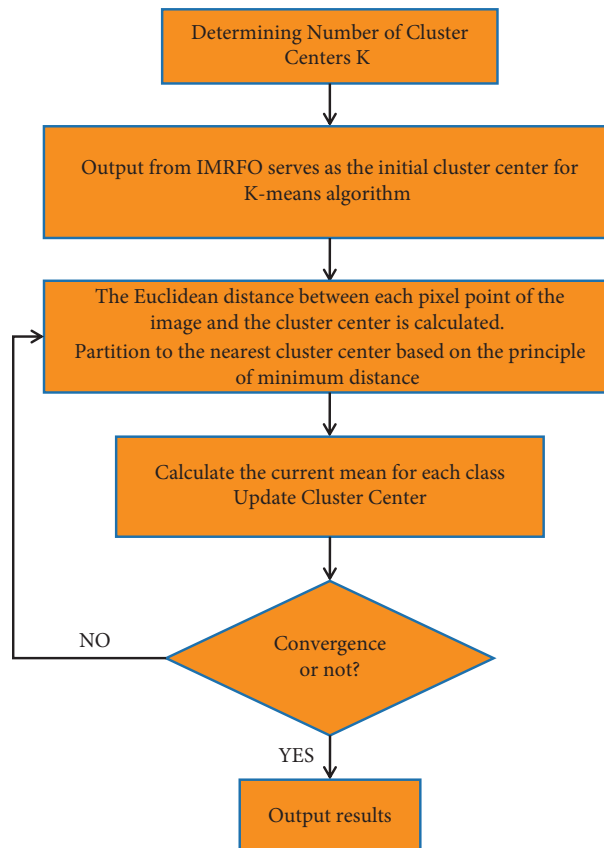
ALGORITHM 1: The framework of the IMRFO.

FIGURE 3: K-Means image segmentation based on IMRFO.

selected to verify its function optimization ability. The specific test function information is shown in Table 1. F1-6 is a unimodal function, F7-11 is a complex multimodal function, and F12 is a fixed-dimensional function. In addition, F1-11 is tested in different dimensions to verify the optimization ability of the algorithm in high-dimensional cases. To prove that IMRFO is competitive, seven algorithms including MRFO, Honey Badger Algorithm (HBA) [45], GWO, PSO, Whale Optimization Algorithm (WOA) [46], Learning Based Optimization (TLBO) [47], and Flower Pollination Algorithm (FPA) [48] are compared. The new cluster intelligence algorithm was proposed by HBA in 2021, while other algorithms are classical ones that have been extensively studied. The number of iterations and population of each algorithm are 500 and 100. In HBA, $O = 6$, $C = 2$; In FPA, the selection probability $p = 0.8$. B_1 and b_2 in IMRFO are 0.2 and 0.8, respectively. The experimental environment is Windows 10 64 bit; the software is matlab2019b; the memory is 16 GB; the processor is Intel(R) Core (TM) i5-10200H CPU @ 2.40 GHz. The average, optimal value, and standard deviation of the results of each algorithm for 30 runs are calculated. If IMRFO is the optimal value, the font is bolded. The optimization results of each algorithm are calculated as shown in Table 2-3.

On the one hand, from Table 2 and Table 3, we can see that IMRFO has obvious advantages in searching ability, and the results are better than other algorithms in each function. The increase of dimension does not reduce the searching ability of IMRFO. On the other hand, among these functions, F1, F6, F8-10, F12, MRFO itself has a good optimization effect and can find the theoretical optimal value, IMRFO also has the same optimization effect, so it can be seen that IMRFO does not weaken the original algorithm's optimization ability. Overall, IMRFO has been effectively improved in stability and accuracy. It can be seen that the introduction of multiple strategies improves the algorithm's optimization ability and reduces the probability of entering a local optimum.

5.2. Statistical Test.

To verify whether IMRFO and the other seven algorithms have significant differences in global optimization, the 30-dimension results of each algorithm are tested. The Wilcoxon rank-sum test is used to find the differences between the two algorithms. Assume H0 : The two algorithms have the same performance. H1 : There is an obvious difference between the two algorithms. Use the P-value of the test results to measure the differences between the two algorithms. When $P < 0.05$, reject H0. It shows that there is a significant difference between the two algorithms. When $P > 0.05$, H0 is accepted, indicating that the two algorithms have the same global optimization performance. To clearly see the differences between these algorithms, we utilize N/A to represent the values of $P > 0.05$. The Wilcoxon test results are shown in Table 4. At the same time, in order to better show the comprehensive optimization ability of IMRFO in the whole test function, the average and variance of each algorithm are Friedman test [49], and the final ranking is calculated to measure the universality of the

algorithm in the 12 test functions. The test results are shown in Table 5.

From Table 4, it can be seen that IMRFO differs significantly from other algorithms. In some functions, MRFO itself has better searching ability, so the difference is not obvious. From Table 5, IMRFO ranks best in the search results of each function, which also indicates that it has a good universality.

5.3. Comparison with Variants of the Algorithm.

To further show the effectiveness and innovation of IMRFO, this paper compares IMRFO with multistrategy serial cuckoo search algorithm (MSSCS) [50], firefly algorithm with courtship learning (FACL) [51], self-adaptive cuckoo search (SACS) [52], and CSsin [53] proposed in recent years. These four algorithms are variants of classical algorithms and have been validated in the CEC test set. The specific parameters of each algorithm are set as follows: In MSCS algorithm, $\alpha = 0.01$, $\beta = 1.5$, $P_a = 0.25$, $C = 0.2$, $PA_{max} = 0.35$, $PA_{min} = 0.25$; in CSsin algorithm, $P_{max} = 0.75$, $P_{max} = 0.25$, $freq = 0.5$; in FACL, $\alpha = 0.01$, $\beta_{min} = 0.2$, $\beta = 1$, $\gamma = 1$. The number of populations and the number of iterations for each algorithm are shown above. Similarly, when IMRFO is the optimal value, font bolding will be applied. The results of each algorithm are shown in Table 6.

From Table 6, it is clear that IMRFO is the best value in F1-4, F6, F9-12, which shows that IMRFO is better than these algorithms in the optimization of these functions. Secondly, the variants of CS have better optimization results, especially in F5 and F7, which have higher accuracy. FACL, as the worst one, has poor optimization results but good stability. Generally speaking, IMRFO has some advantages in function optimization, which verifies the effectiveness and innovation of the algorithm.

5.4. Convergence Analysis.

In order to clearly see the optimization and convergence effect of each algorithm in each function, the average convergence diagram of each algorithm is given as shown in Figure 4.

From Figure 4, it can be seen that IMRFO has a good convergence effect and can find the most accurate solution quickly, especially in the functions of F1-4, F6, F11. It can be seen that the flexible search mechanism enables the algorithm to find the best solution quickly in the optimization process.

5.5. Ablation Experiment.

In order to verify the validity and feasibility of the three combinations of strategies, the combinations of strategies are experimented with to find the better one. In this paper, the algorithm of combining Lévy flight with GWL is recorded as MRFO-I, the algorithm of combining Lévy flight with PSO learning thought strategy is recorded as MRFO-II, while the algorithm of combining PSO thought with GWL is recorded as MRFO-III. Besides, the algorithm using Lévy flight alone is recorded as MRFO-IV, the algorithm using GWL alone as MRFO-V, and the algorithm utilizing PSO alone as MRFO-VI. The

TABLE 1: Test function information table.

F	Dim	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} x_i$	30/100	[−10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30/100	[−100,100]	0
$F_4(x) = \max_i\{|x_i|, 1 \le i \le n\}$	30/100	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/100	[−30,30]	0
$F_6(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0,1)$	30/100	[−1.28,1.28]	0
$F_7(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right)$	30/100	[−500,500]	−418.9829n
$F_8(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30/100	[−5.12,5.12]	0
$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30/100	[−32,32]	0
$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30/100	[−600,600]	0
$F_{11}(x) = (\pi/n)\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i,10,100,4)$			
$y_i = 1 + (x_i + 1/4)$	30/100	[−50,50]	0
$u(x_i,a,k,m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a, \\ k(-x_i - a)^m & x_i < -a. \end{cases}$			
$F_{12}(x) = \left(0.002 + \sum_{i=1}^{25}\left(1/i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6\right)^{-1}\right)^{-1}$ where $a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{pmatrix}$	2	[−65.536,65.536]	0.998

Table 2: Performance comparison table of each algorithm

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	0	0	**0**
	MRFO	0	0	0
	HBA	$9.3799E-166$	$1.1587E-160$	$4.2983E-160$
	WOA	$3.1181E-104$	$1.0841E-95$	$4.5034E-95$
	TLBO	$4.0530E-86$	$2.9876E-85$	$2.1878E-85$
	FPA	$3.4196E+04$	$5.6323E+04$	$9.3090E+03$
	GWO	$2.97266E-42$	$9.97975E-41$	$2.31134E-40$
	PSO	$1.35628E-12$	$7.30343E-11$	$1.53846E-10$
$F_2(x)$	IMRFO	**0**	**$4.8161E-277$**	**0**
	MRFO	$4.2592E-237$	$1.5644E-229$	0
	HBA	$1.0045E-87$	$3.0642E-85$	$8.9871E-85$
	WOA	$5.1550E-63$	$8.1673E-58$	$2.9729E-57$
	TLBO	$3.0778E-43$	$1.2493E-42$	$7.1358E-43$
	FPA	$4.4336E+07$	$7.9783E+10$	$1.3371E+11$
	GWO	$1.00188E-24$	$5.16621E-24$	$3.79647E-24$
	PSO	$1.9394E-06$	$1.2453E-04$	$3.7943E-04$
$F_3(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	$4.2858E-192$	0
	HBA	$7.5465E-129$	$1.8442E-117$	$6.9103E-117$
	WOA	$4.7217E+03$	$1.4248E+04$	$6.9878E+03$
	TLBO	$6.0458E-17$	$1.7598E-15$	$1.8686E-15$
	FPA	$6.4669E+04$	$9.4629E+04$	$1.9121E+04$
	GWO	$1.2723E-15$	$1.6522E-11$	$5.6380E-11$
	PSO	2.4597	6.8764	4.6664
$F_4(X)$	IMRFO	**0**	**$3.6613E-283$**	**0**
	MRFO	$1.8994E-233$	$1.2937E-223$	0
	HBA	$6.4810E-72$	$3.6559E-68$	$8.7750E-68$
	WOA	$1.3789E-07$	$2.9967E+01$	$3.0998E+01$
	TLBO	$7.54383E-35$	$2.2303E-34$	$1.1471E-34$
	FPA	$7.5982E+01$	$8.3319E+01$	$3.0315E+00$
	GWO	$1.0249E-11$	$1.3528E-10$	$1.3630E-10$
	PSO	$1.0009E+01$	$1.0221E+02$	$5.1654E+01$
$F_5(X)$	IMRFO	$1.8761E+01$	$1.9089E+01$	$4.2393E-01$
	MRFO	$1.9091E+01$	$2.0074E+01$	$3.9236E-01$
	HBA	$1.9310E+01$	$2.1199E+01$	$7.6666E-01$
	WOA	$2.6313E+01$	$2.6797E+01$	$2.0942E-01$
	TLBO	$1.6702E+01$	$1.8846E+01$	1.0315
	FPA	$1.0583E+08$	$1.7478E+08$	$3.8136E+07$
	GWO	$4.5186E+01$	$4.6672E+01$	$7.8488E-01$
	PSO	89.4507	234.8606	104.6002
$F_6(X)$	IMRFO	**$1.4166E-06$**	**$2.9500E-05$**	**$2.0356E-05$**
	MRFO	$3.1973E-06$	$3.8448E-05$	$4.0550E-05$
	HBA	$1.6931E-05$	$1.1151E-04$	$8.9566E-05$
	WOA	$3.9351E-05$	$9.5517E-04$	$9.4421E-04$
	TLBO	$4.4041E-04$	$7.9732E-04$	$2.1545E-04$
	FPA	$5.1790E+01$	$9.3728E+01$	$1.9183E+01$
	GWO	$1.8171E-04$	$5.6482E-04$	$3.1680E-04$
	PSO	$7.1930E-03$	$1.8163E-02$	$7.2361E-03$
$F_7(X)$	IMRFO	$-1.0536E+04$	$-1.0050E+04$	$4.2525E+02$
	MRFO	$-9.6874E+03$	$-9.0741E+03$	$4.3594E+02$
	HBA	$-1.0990E+04$	$-9.4427E+03$	$9.4018E+02$
	WOA	-12569.4699	-11724.03485	1302.326078
	TLBO	$-9.2643E+03$	$-7.3757E+03$	$1.1529E+03$
	FPA	$-4.0064E+03$	$-2.9596E+03$	$3.8588E+02$
	GWO	$-8.2622E+03$	$-6.3621E+03$	$7.0903E+02$
	PSO	$-8.3241E+03$	$-6.8450E+03$	$7.5268E+02$

TABLE 2: Continued.

F	Algorithm	Best	Mean	Std
$F_8(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	$1.8948E-15$	$1.0378E-14$
	TLBO	0	6.3284	4.7427
	FPA	$3.3516E+02$	$3.9820E+02$	$2.9755E+01$
	GWO	0	$4.7528E-01$	1.4588
	PSO	0	2.0630	1.3861
$F_9(X)$	IMRFO	**$8.8818E-16$**	**$8.8818E-16$**	**0**
	MRFO	$8.8818E-16$	$8.8818E-16$	0
	HBA	$8.8818E-16$	$8.8818E-16$	0
	WOA	$8.8818E-16$	$4.0856E-15$	$2.6960E-15$
	TLBO	$4.4409E-15$	$6.3357E-15$	0
	FPA	$1.9290E+01$	$2.0337E+01$	$2.7610E-01$
	GWO	$1.8652E-14$	$2.7652E-14$	$3.8165E-15$
	PSO	$1.7249E-06$	$1.9663E-01$	$4.5356E-01$
$F_{10}(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	$1.9723E-03$	$3.3266E-03$
	TLBO	0	0	0
	FPA	$2.7736E-01$	$1.2829E+00$	$6.1301E-01$
	GWO	0	$2.4654E-03$	$3.5462E-03$
	PSO	0	$1.9723E-03$	$3.3266E-03$
$F_{11}(X)$	IMRFO	**$7.3400E-20$**	**$1.2392E-18$**	**$1.2894E-18$**
	MRFO	$4.04531E-06$	$1.53518E-05$	$8.77191E-06$
	HBA	$3.8903E-10$	$6.7898E-09$	$1.1908E-08$
	WOA	$2.7893E-03$	$4.9054E-03$	$1.5734E-03$
	TLBO	$5.7527E-05$	$2.2424E-03$	$5.7778E-03$
	FPA	$5.1790E+07$	$3.7356E+08$	$1.3609E+08$
	GWO	$1.4152E-06$	$1.4681E-02$	$9.2775E-03$
	PSO	$6.4838E-14$	$8.6436E-02$	$2.5276E-01$
$F_{12}(X)$	IMRFO	**$9.9800E-01$**	**$9.9800E-01$**	**0**
	MRFO	$9.9800E-01$	$9.9800E-01$	0
	HBA	$9.9800E-01$	$9.9800E-01$	0
	WOA	$9.9800E-01$	1.2626	$6.8599E-01$
	TLBO	$9.9800E-01$	$9.9800E-01$	0
	FPA	1.9995	$1.0262E+01$	7.6364
	GWO	$9.9800E-01$	$1.7255E+00$	$9.7247E-01$
	PSO	$9.9800E-01$	$1.1305E+00$	$3.4368E-01$

The optimal value is shown in bold.

TABLE 3: Table of results for each algorithm (100 DIMENSIONS).

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	8.9040E-147	4.6592E-141	1.2550E-140
	WOA	1.0469E-102	7.2231E-93	2.5499E-92
	TLBO	1.6114E-75	1.6114E-75	1.1585E-75
	FPA	1.3848 E+05	2.2811 E+05	4.4746 E+04
	GWO	5.0821E-18	2.6308E-17	2.1090E-17
	PSO	1.0669	1.2459 E+01	3.0029 E+01
$F_2(x)$	IMRFO	**2.0908E-305**	**4.0935E-281**	**0**
	MRFO	8.2591E-234	8.7997E-228	0
	HBA	3.2693E-78	1.4848E-75	2.1169E-75
	WOA	3.97136E-62	8.1673E-58	4.04975E-55
	TLBO	1.58996E-38	1.11006E-55	1.64669E-38
	FPA	2.5777 E+27	4.8007 E+45	1.3669 E+46
	GWO	1.1188E-23	1.0071E-22	8.1329E-23
	PSO	1.6825E-06	4.0865E-05	1.3518E-04

TABLE 3: Continued.

F	Algorithm	Best	Mean	Std
$F_3(X)$	IMRFO	**0**	**0**	0
	MRFO	0	4.2858E-192	0
	HBA	5.8574E-106	6.5344E-94	3.2931E-93
	WOA	4.5880 E+05	6.7389 E+05	1.1901E+05
	TLBO	5.7047E-06	1.4581E-04	1.5072E-04
	FPA	5.9828 E+05	9.8255 E+05	2.2421 E+05
	GWO	4.3996E-01	1.1814 E+01	1.3585 E+01
	PSO	4.1855 E+03	6.8320 E+03	7.6320 E+03
$F_4(X)$	IMRFO	**2.0908E-305**	**4.0728E-281**	**0**
	MRFO	1.0891E-224	6.0212E-219	0
	HBA	4.1580E-49	6.4459E-47	1.2824E-46
	WOA	9.9712E-04	7.4068 E+01	2.9309 E+01
	TLBO	2.64171E-30	4.63468E-30	1.35657E-30
	FPA	8.7604 E+01	9.3681 E+01	2.0880
	GWO	2.0983E-03	5.2094E-02	8.6554E-02
	PSO	7.3783	9.5754	1.2668
$F_5(X)$	IMRFO	**9.0554 E+01**	9.1195 E+01	7.3432E-01
	MRFO	9.0823 E+01	9.2361 E+01	6.6140E-01
	HBA	9.1619 E+01	9.4762 E+01	1.8652 E+00
	WOA	9.6679 E+01	9.7258 E+01	3.0684E-01
	TLBO	9.1198 E+01	9.3064 E+01	9.5488E-01
	FPA	4.0625 E+08	9.3161 E+08	2.3879 E+08
	GWO	9.5679 E+01	9.6973 E+01	9.2081E-01
	PSO	6.5109 E+02	3.2217 E+03	6.7489 E+03
$F_6(X)$	IMRFO	**1.4328E-06**	**6.6842E-05**	4.9030E-05
	MRFO	2.0781E-06	4.9605E-05	4.3316E-05
	HBA	2.3868E-05	1.4383E-04	1.2534E-04
	WOA	8.5089E-06	1.3208E-03	1.3226E-03
	TLBO	5.1751E-04	1.1234E-03	3.2954E-04
	FPA	6.1163 E+02	1.4259 E+03	3.5171 E+02
	GWO	9.4824E-04	2.6521E-03	1.1372E-03
	PSO	4.9985E-01	1.1220 E+00	3.6320E-01
$F_7(X)$	IMRFO	-2.92 E+04	-2.57 E+04	1.67 E+03
	MRFO	-2.7791 E+04	-2.5142 E+04	1.1769 E+03
	HBA	-3.2530 E+04	-2.6092 E+04	3.5550 E+03
	WOA	-4.1898 E+04	-3.9015 E+04	3.7697 E+03
	TLBO	-2.7940 E+04	-1.6598 E+04	5.1640 E+03
	FPA	-6.5597 E+03	-5.3592 E+03	6.0949 E+02
	GWO	-2.2404 E+04	-1.6607 E+04	2.4372 E+03
	PSO	-2.3436 E+04	-2.0057 E+04	2.2771 E+03
$F_8(X)$	IMRFO	**0**	**0**	0
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	3.7896E-15	2.0756E-14
	TLBO	0	6.3284	4.7427
	FPA	1.3434 E+03	1.5311 E+03	8.6281 E+01
	GWO	0	1.3971	2.2329
	PSO	2.8854 E+01	4.8456 E+01	1.1153 E+01
$F_9(X)$	IMRFO	**8.8818E-16**	**8.8818E-16**	**0**
	MRFO	8.8818E-16	8.8818E-16	0
	HBA	8.8818E-16	8.8818E-16	0
	WOA	8.8818E-16	4.3225E-15	2.8731E-15
	TLBO	7.9936E-15	7.9936E-15	0
	FPA	1.9841 E+01	2.0520 E+01	3.3562E-01
	GWO	2.2204E-14	3.0257E-14	3.7242E-15
	PSO	1.3320E-06	1.4424E-01	4.4254E-01

TABLE 3: Continued.

F	Algorithm	Best	Mean	Std
	IMRFO	**0**	**0**	0
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	0	0
$F_{10}(X)$	TLBO	0	0	0
	FPA	1.3815 E+03	2.0376 E+03	3.4470 E+02
	GWO	0	1.8958E-03	3.1773E-03
	PSO	0	1.3192E-03	2.8042E-03
	IMRFO	**2.8193E-06**	**6.4940E-06**	**3.2213E-06**
	MRFO	4.6460E-06	1.09637E-05	6.3630E-06
	HBA	1.9642E-02	3.5856E-02	1.0354E-02
	WOA	1.8029E-03	7.0382E-03	9.1277E-03
$F_{11}(X)$	TLBO	7.3698E-05	2.3727E-03	5.8937E-03
	FPA	7.4824 E+08	2.1393 E+09	7.0466 E+08
	GWO	2.1287E-06	1.5932E-02	9.1126E-03
	PSO	2.3297E-13	8.2984E-02	1.7745E-01
	IMRFO	**9.9800E-01**	**9.9800E-01**	**0**
	MRFO	9.9800E-01	9.9800E-01	0
	HBA	9.9800E-01	9.9800E-01	0
	WOA	9.9800E-01	1.2626	6.8599E-01
$F_{12}(X)$	TLBO	9.9800E-01	9.9800E-01	0
	FPA	1.9995	1.0262 E+01	7.6364
	GWO	9.9800E-01	1.7255 E+00	9.7247E-01
	PSO	9.9800E-01	1.1305 E+00	3.4368E-01

Bold font is to clearly see the advantages of the algorithm. The best indicators of IMRFO have been bold in this paper.

TABLE 4: Wilcoxon rank-sum test p-value.

F	MRFO	FPA	GWO	HBA	PSO	TLBO	WOA
F1	N/A	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F2	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F3	N/A	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F4	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F5	1.46E-09	1.79E-11	1.79E-11	4.94E-11	3.53E-10	N/A	1.79E-11
F6	N/A	2.91E-11	2.91E-11	5.07E-07	2.91E-11	2.91E-11	2.29E-10
F7	7.61E-09	1.44E-11	1.44E-11	3.31E-03	1.44E-11	1.97E-11	7.40E-07
F8	N/A	1.21E-12	8.15E-02	N/A	4.24E-12	1.66E-11	3.34E-01
F9	N/A	1.21E-12	6.59E-13	N/A	1.21E-12	4.63E-13	1.02E-07
F10	N/A	1.21E-12	6.62E-04	N/A	1.35E-03	N/A	2.79E-03
F11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F12	5.46E-03	1.14E-11	1.14E-11	4.95E-10	6.79E-02	2.50E-05	1.14E-11

TABLE 5: Friedman test ranking table.

Algorithms	IMRFO	MRFO	HBA	WOA	TLBO	FPA	GWO	PSO
Rank	2.0208	2.2708	2.8333	5.0000	4.4583	7.5417	5.4167	6.4583

experimental parameters are consistent with those above. The test function dimension is 30. If IMRFO is the optimal value, the font is bolded. The experimental results are shown in Table 7.

As can be seen from Table 7, IMRFO is the best performer of all variants, and the criteria on each function are the best. IMRFO search accuracy is better than other algorithms and the difference is significant especially in F2, F4, F11. Therefore, it can be seen that the integration of multiple strategies is important, and the validity and feasibility of IMRFO are verified.

5.6. Time Complexity Analysis. Time complexity is an important measure of an algorithm. In order to show an effective improvement, it is necessary to balance the searchability and time complexity of the algorithm. The basic MRFO consists of only three phases, chain feeding, spiral feeding, and empty feeding, where chain feeding and spiral feeding are in the same cycle. Set the population number to N and the maximum number of iterations to T. The dimension is D, so the time complexity of MRFO can be summarized as follows. Macroscopically, the time complexity of swarm intelligence algorithms is the product of

TABLE 6: Comparison with variants of the algorithm

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	**0**	**0**	**0**
	FACL	$2.6020E+02$	$2.6020E+02$	$4.5704E-13$
	MSSCS	$2.6586E-16$	$1.2320E-14$	$2.2109E-14$
	SACS	$5.8993E-17$	$2.1975E-15$	$2.2919E-15$
	CSsin	$8.1279E-17$	$1.8682E-15$	$2.2000E-15$
$F_2(x)$	IMRFO	**0**	**$4.8161E-277$**	**0**
	FACL	$1.3615E+01$	$1.3615E+01$	$1.7853E-15$
	MSSCS	$2.4907E-09$	$1.0918E-08$	$9.6029E-09$
	SACS	$8.1604E-10$	$3.6463E-09$	$2.3128E-09$
	CSsin	$8.7299E-10$	$3.3430E-09$	$1.7651E-09$
$F_3(X)$	IMRFO	**0**	**0**	**0**
	FACL	$2.3244E+03$	$2.3244E+03$	$2.7422E-12$
	MSSCS	$2.1933E-02$	$6.6588E-01$	$1.2328E+00$
	SACS	$2.0617E-13$	$4.2875E-10$	$7.7897E-10$
	CSsin	$3.4984E-13$	$3.5258E-09$	$1.8741E-08$
$F_4(X)$	IMRFO	**0**	**$3.6613E-283$**	**0**
	FACL	$1.5622E+01$	$1.5622E+01$	$1.4282E-14$
	MSSCS	$9.5018E-05$	$2.8057E-03$	$7.0418E-03$
	SACS	$2.5496E-15$	$2.8163E-12$	$8.9570E-12$
	CSsin	$1.1423E-15$	$1.3641E-12$	$4.9902E-12$
$F_5(X)$	IMRFO	$1.8761E+01$	$1.9089E+01$	$4.2393E-01$
	FACL	$6.2978E+04$	$6.2978E+04$	$7.3126E-11$
	MSSCS	$2.1186E+01$	$2.6991E+01$	$1.5287E+01$
	SACS	$9.5265E-03$	$1.6347E+01$	$1.0434E+01$
	CSsin	$3.1439E+00$	$1.4000E+01$	$4.5249E+00$
$F_6(X)$	IMRFO	**$1.4166E-06$**	**$2.9500E-05$**	**$2.0356E-05$**
	FACL	$4.9638E-01$	$2.6540E+00$	$3.6836E+00$
	MSSCS	$1.0931E-03$	$3.7296E-03$	$1.7917E-03$
	SACS	$2.7015E-04$	$1.2413E-03$	$6.5230E-04$
	CSsin	$2.5773E-04$	$1.2047E-03$	$6.2553E-04$
$F_7(X)$	IMRFO	$-1.0536E+04$	$-1.0050E+04$	$4.2525E+02$
	FACL	$-3.2588E+03$	$-3.2588E+03$	$3.6563E-12$
	MSSCS	$-1.2409E+04$	$-1.2023E+04$	$1.7642E+02$
	SACS	$-1.2337E+04$	$-1.2040E+04$	$1.6571E+02$
	CSsin	$-1.2197E+04$	$-1.1773E+04$	$2.2010E+02$
$F_8(X)$	IMRFO	**0**	**0**	**0**
	FACL	$1.3469E+02$	$1.3469E+02$	$1.9995E-13$
	MSSCS	$1.3682E+02$	$1.3682E+02$	$2.2852E-13$
	SACS	$1.3558E+01$	$1.9318E+01$	$3.2639E+00$
	CSsin	$1.3082E+01$	$1.7173E+01$	$2.4068E+00$
$F_9(X)$	IMRFO	**$8.8818E-16$**	**$8.8818E-16$**	**0**
	FACL	$9.5803E+00$	$9.5803E+00$	$1.4282E-14$
	MSSCS	$7.8366E-09$	$3.3808E-08$	$2.3225E-08$
	SACS	$6.6614E-09$	$3.5846E-08$	$2.6071E-08$
	CSsin	$9.4992E-09$	$4.9013E-08$	$3.5110E-08$
$F_{10}(X)$	IMRFO	**0**	**0**	**0**
	FACL	$1.6979E+01$	$1.6979E+01$	$2.4994E-14$
	MSSCS	$6.6613E-16$	$8.2473E-04$	$4.4988E-03$
	SACS	$1.6653E-15$	$3.2272E-03$	$9.0416E-03$
	CSsin	$6.6613E-16$	$1.8055E-03$	$8.5946E-03$
$F_{11}(X)$	IMRFO	**$7.3400E-20$**	**$1.2392E-18$**	**$1.2894E-18$**
	FACL	$8.7100E+00$	$8.7100E+00$	$1.4282E-14$
	MSSCS	$1.5139E-07$	$5.4848E-07$	$3.2760E-07$
	SACS	$2.2637E-07$	$4.8725E-07$	$3.3696E-07$
	CSsin	$1.7143E-07$	$5.0219E-07$	$3.9414E-07$
$F_{12}(X)$	IMRFO	**$9.9800E-01$**	**$9.9800E-01$**	**0**
	FACL	$1.9920E+00$	$1.9920E+00$	$4.4633E-15$
	MSSCS	—	—	—
	SACS	$9.9800E-01$	$9.9800E-01$	0
	CSsin	$9.9800E-01$	$9.9800E-01$	0

The optimal value is shown in bold.
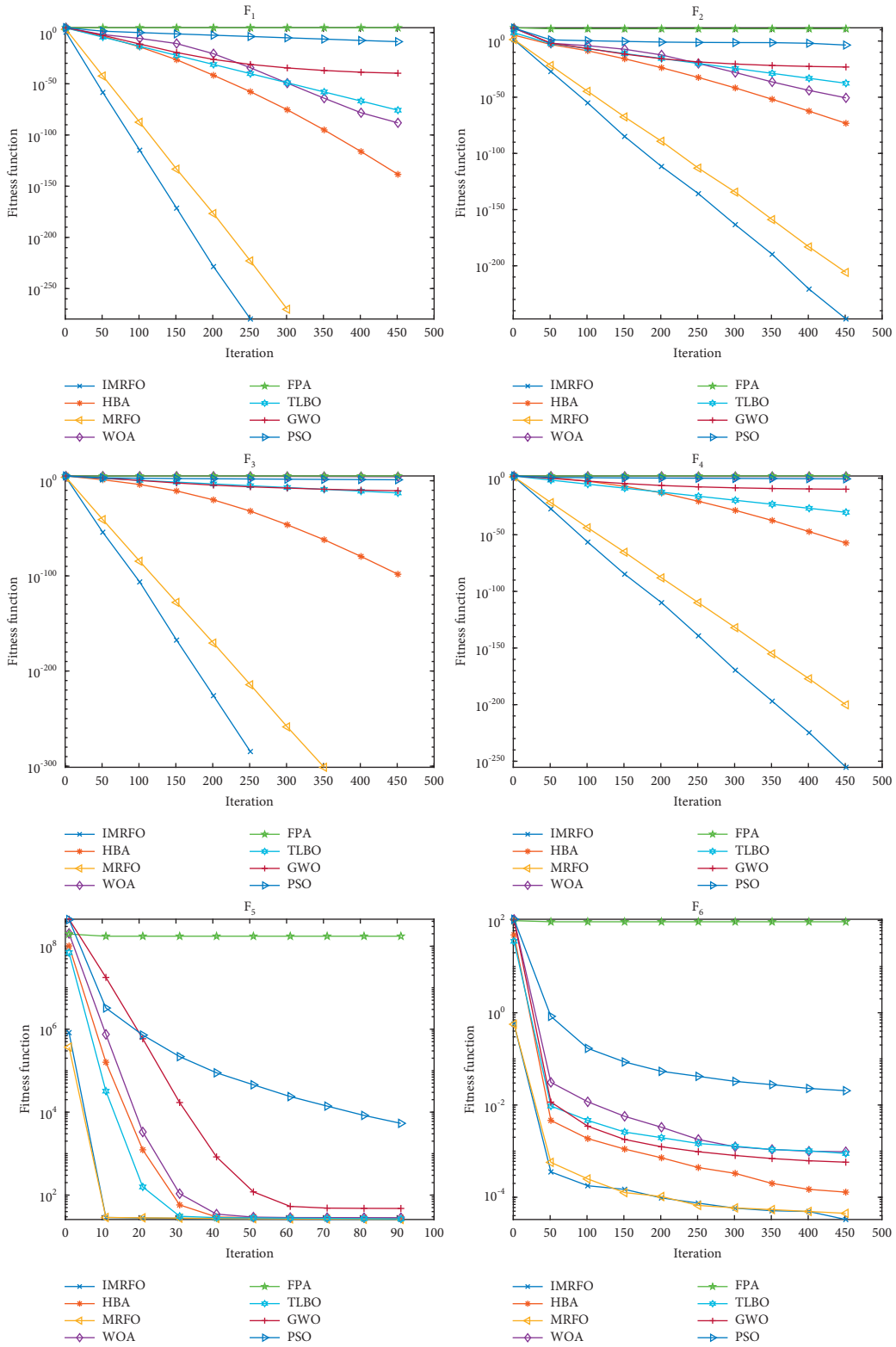
(a)

FIGURE 4: Continued.
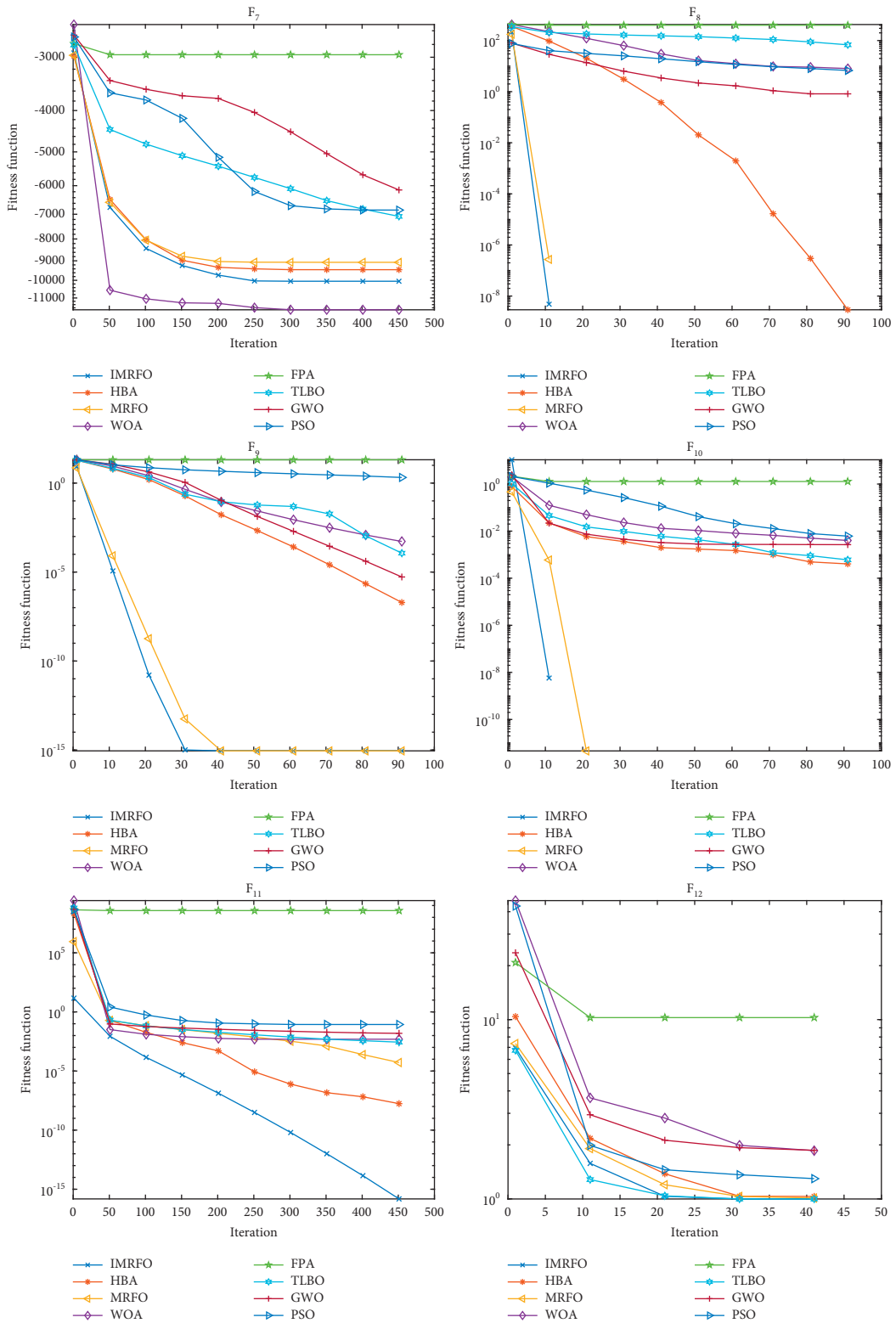
Figure 4: Average convergence of each algorithm.

TABLE 7: Test table for each combination algorithm.

F	Algorithm	Best	Mean	Std
	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
$F_1(x)$	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
	IMRFO	**0**	**4.8161E-277**	**0**
	MRFO-I	2.2366E-239	2.3660E-230	0
	MRFO-II	1.4788E-239	3.6180E-231	0
$F_2(x)$	MRFO-III	6.7519E-240	2.1114E-231	0
	MRFO-IV	4.9710E-237	1.7202E-229	0
	MRFO-V	3.4101E-236	1.0691E-229	0
	MRFO-VI	6.6058E-239	8.1070E-230	0
	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
$F_3(X)$	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
	IMRFO	**0**	**3.6613E-283**	**0**
	MRFO-I	7.1957E-231	9.1743E-224	0
	MRFO-II	5.5959E-232	2.5385E-224	0
$F_4(X)$	MRFO-III	2.2845E-232	2.3915E-224	0
	MRFO-IV	1.7079E-231	6.1960E-224	0
	MRFO-V	3.1285E-233	3.7719E-225	2.0880
	MRFO-VI	3.3266E-233	2.8006E-224	8.6554E-02
	IMRFO	**1.8761 E+01**	**1.9089 E+01**	4.2393E-01
	MRFO-I	1.9160 E+01	2.0228 E+01	4.7486E-01
	MRFO-II	1.9205 E+01	1.9904 E+01	4.1531E-01
$F_5(X)$	MRFO-III	1.9050 E+01	2.0040 E+01	4.5484E-01
	MRFO-IV	1.9054 E+01	1.9923 E+01	4.8479E-01
	MRFO-V	1.8959 E+01	2.0027 E+01	5.1658E-01
	MRFO-VI	1.8883 E+01	1.9994 E+01	6.7667E-01
	IMRFO	**1.4166E-06**	**2.9500E-05**	**2.0356E-05**
	MRFO-I	3.3254E-06	4.0901E-05	2.9878E-05
	MRFO-II	1.7086E-06	4.2369E-05	3.8592E-05
$F_6(X)$	MRFO-III	2.9884E-06	4.3133E-05	4.4862E-05
	MRFO-IV	5.3342E-06	5.1792E-05	5.3359E-05
	MRFO-V	8.3930E-06	6.0626E-05	4.2128E-05
	MRFO-VI	4.8785E-06	5.3014E-05	4.7227E-05
	IMRFO	**-1.0536 E+04**	**-1.0050 E+04**	**4.2525 E+02**
	MRFO-I	-9.7467 E+03	-8.7240 E+03	6.4405 E+02
	MRFO-II	-9.9243 E+03	-8.7951 E+03	6.3557 E+02
$F_7(X)$	MRFO-III	-9.8848 E+03	-8.8964 E+03	5.5148 E+02
	MRFO-IV	-1.0102 E+04	-8.7470 E+03	6.3365 E+02
	MRFO-V	-1.0359 E+04	-8.9240 E+03	7.0323 E+02
	MRFO-VI	-1.0082 E+04	-9.0438 E+03	5.6962 E+02
	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
$F_8(X)$	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0

F	Algorithm	Best	Mean	Std
	IMRFO	**8.8818E-16**	**8.8818E-16**	**0**
	MRFO-I	8.8818E-16	8.8818E-16	0
	MRFO-II	8.8818E-16	8.8818E-16	0
$F_9(X)$	MRFO-III	8.8818E-16	8.8818E-16	0
	MRFO-IV	8.8818E-16	8.8818E-16	0
	MRFO-V	8.8818E-16	8.8818E-16	0
	MRFO-VI	8.8818E-16	8.8818E-16	0
	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
$F_{10}(X)$	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
	IMRFO	**7.3400E-20**	1.2392E-18	1.2894E-18
	MRFO-I	1.9606E-19	1.5295E-18	1.7575E-18
	MRFO-II	1.8111E-19	1.4964E-18	1.1884E-18
$F_{11}(X)$	MRFO-III	1.2951E-19	2.0136E-18	2.1218E-18
	MRFO-IV	3.8758E-19	2.7853E-18	4.2987E-18
	MRFO-V	1.1870E-19	1.1469E-18	1.0703E-18
	MRFO-VI	9.2691E-20	1.8700E-18	2.1327E-18
	IMRFO	**9.9800E-01**	**9.9800E-01**	**0**
	MRFO-I	9.9800E-01	9.9800E-01	0
	MRFO-II	9.9800E-01	9.9800E-01	0
$F_{12}(X)$	MRFO-III	9.9800E-01	9.9800E-01	0
	MRFO-IV	9.9800E-01	9.9800E-01	0
	MRFO-V	9.9800E-01	9.9800E-01	0
	MRFO-VI	9.9800E-01	9.9800E-01	0

Bold font is to clearly see the advantages of the algorithm.

population number, iteration number, and dimension. Therefore, the time complexity of IMRFO is O (TND), just like other algorithms.

Microscopically, MRFO can be calculated as follows:

$$O(\text{IMRFO}) = O(T(O(\text{cyclone foraging} + \text{chain foraging})$$
$$+ O(\text{somersault foraging})))$$
$$= O(T(ND + ND)) = O(TND). \tag{11}$$

$$O(\text{MRFO}) = O(T(O(\text{cyclone foraging} + \text{chain foraging}) + O(t1) + O(\text{somersault foraging}) + O(t_2) + O(t_3)))$$
$$= O(TND). \tag{12}$$

Therefore, it can be seen that the time complexity of IMRFO has not changed fundamentally. A small increase in the number of iterations can be ignored. These increases will be of great significance if the optimization capability of the algorithm is effectively improved.

6. Image Segmentation Experiments

At present, image processing has been applied in many fields, and the image on land has been well developed, but it still has research value in underwater images. So in order to show the research value, eight underwater images

Set the calculation time of introducing RWL to be t1, the calculation time of introducing Lévy flight to be t_2, the calculation time of using two learning factors to be t_3, and the other calculations are ignored.

IMRFO can be summarized as follows:

are selected as test images. From the literature [54], select PSO, DPSO, sparrow search algorithm (SSA) [55], Modified sparrow search algorithm (MSSA) [56], ABC, MRFO, WOA, TLBO, FPA, IMRFO 9 algorithms optimize K-means algorithm and traditional K-means algorithm for image segmentation. MSSA is a newly proposed K-means based algorithm, and other algorithms have been successfully applied to image segmentation problems in recent years. Because the K-means clustering algorithm has a strong dependence on K values, improper selection of K values will have a great impact on the results, and K-means clustering algorithm has a strong dependence

on K values. Set the value of *k* to 3 to avoid interference from unrelated factors. The general parameters of the algorithm are population size of 30 and the maximum number of iterations of 100. Each algorithm divides the image as shown in Figure 5 and Figure 6. The first line in Figure 5 and Figure 6 represents the original image, and each subsequent line represents the segmentation effect of each algorithm.

It is impossible to see the difference between each algorithm in image segmentation by human eyes. Therefore, three commonly used image segmentation metrics, PSNR, SSIM, and FSIM, are selected to measure the quality of each algorithm.

Peak Signal-to-Noise Ratio (PSNR) is mainly used to measure the difference between the segmented image and the original image. The formula is as follows [57]:

$$PSNR = 20.\log_{10}\left(\frac{255}{RMSE}\right),$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}(I(i,j) - Seg(i,j))^2}{M \times Q}}. \tag{13}$$

In formula (12) and (13), RMSE represents the root mean square error of the pixels; $M \times Q$ represents the size of the image; $I(i,j)$ represents the pixel gray value of the original image; $Seg(i,j)$ represents the pixel gray value of the segmented image. The larger the PSNR value, the better the segmented image quality. Generally speaking, PSNR higher than 40 dB indicates excellent image quality (indicating that it is very close to the original image). At 30–40db, it usually indicates that the image quality is good (indicating that the distortion is perceptible but acceptable).

Structural Similarity (SSIM) is used to measure the similarity between the original image and the segmented image. The larger the SSIM, the better the segmented result. SSIM is defined as follows [58]:

$$SSIM = \frac{(2\mu_I\mu_{seg} + c_1)(2\sigma_{I,seg} + c_2)}{(\mu_I^2 + \mu_{seg}^2 + c_1)(\sigma_I^2 + \sigma_{seg}^2 + c_2)}. \tag{14}$$

In formula (14), μ_I and μ_{seg} represent the average value of the original image and the segmented image; σ_I and σ_{seg} represent the standard deviation of the original image and the segmented image, respectively.; $\sigma_{I,seg}$ represents the covariance between the original image and the segmented image; c_1, c_2 are constantly used to ensure stability. SSIM value range [0,1]. The larger the value, the smaller the image distortion.

Feature similarity index mersure (FSIM) is a measure of the characteristic similarity between the original image and the quality of the segmentation, used to evaluate local structure and provide contrast information. The value range of FSIM is [0,1], and the closer the value is to 1, the better the segmentation effect. FSIM is defined as follows [59]:

$$FSIM = \frac{\sum_{l\in\Omega}S_L(X)PC_m(X)}{\sum_{l\in\Omega}PC_m(X)},$$

$$SL(X) = S_{PC}(X)S_G(X),$$

$$S_{PC}(X) = \frac{2PC_1(X)PC_2(X) + T_1}{PC_1^2(X)PC_2^2(X) + T_1},$$

$$PC(X) = \frac{E(X)}{(\varepsilon + \sum_m A_n(X))}. \tag{15}$$

$$G = \sqrt{G_x^2 + G_y^2},$$

$$S_G(X) = \frac{2G_1(X)G_2(X) + T_2}{G_1^2(X)G_2^2(X) + T_2},$$

In the above formula, Ω is all the pixel regions of the original image; $S_L(X)$ is the similarity score; $PC_m(X)$ is the phase consistency measure; T_1 and T_2 are constants; G is the gradient descent; $E(x)$ is the response vector size at position X and the scale is *n*; ε is a very small value; $An(X)$ is the local size at scale *n*.

Run each algorithm 10 times, and the average and average running time of the partitioned metrics are shown in Table 8.

Simply from the naked eye, the image after IMRFO segmentation in Figures 5 and Figure 6 is clearer. Some algorithms have a rough segmentation effect and have appeared a blurry phenomenon. From Table 7, it can be seen that the segmentation index of IMRFO has a greater advantage, especially in test01 and test03-08, where more than two indexes are optimal. For example, the FSIM index in test07 reaches 0.97, SSIM in test08 reaches 0.87, which has a significant advantage over other algorithms. When the performance indicator is not optimal, IMRFO is still close to the optimal value. For example, in test01, the SSIM index of WOA is 0.7488, while that of IMRFO is 0.7479, which is close to the optimal value. In test06, the PSNR of ABC was 43.3715, and that of IMRFO was 43.1626. Therefore, both subjective visual effect and measurement result of IMRFO is better than other algorithms, which can prove a good segmentation effect. It also indirectly proves the good search performance of IMRFO, solves the problem that MRFO is easy to fall into local optimal solution and K-means has the disadvantage of being sensitive to the initial clustering center, which results in an excellent initial clustering center and further improves the image segmentation quality. On the other hand, the running time of K-means segmentation is the least, but the quality is the worst. The operation of other algorithms is large and the effect is obvious. The IMRFO does have a time disadvantage, to be expected, as it takes more time to accurately scan the solution in space.
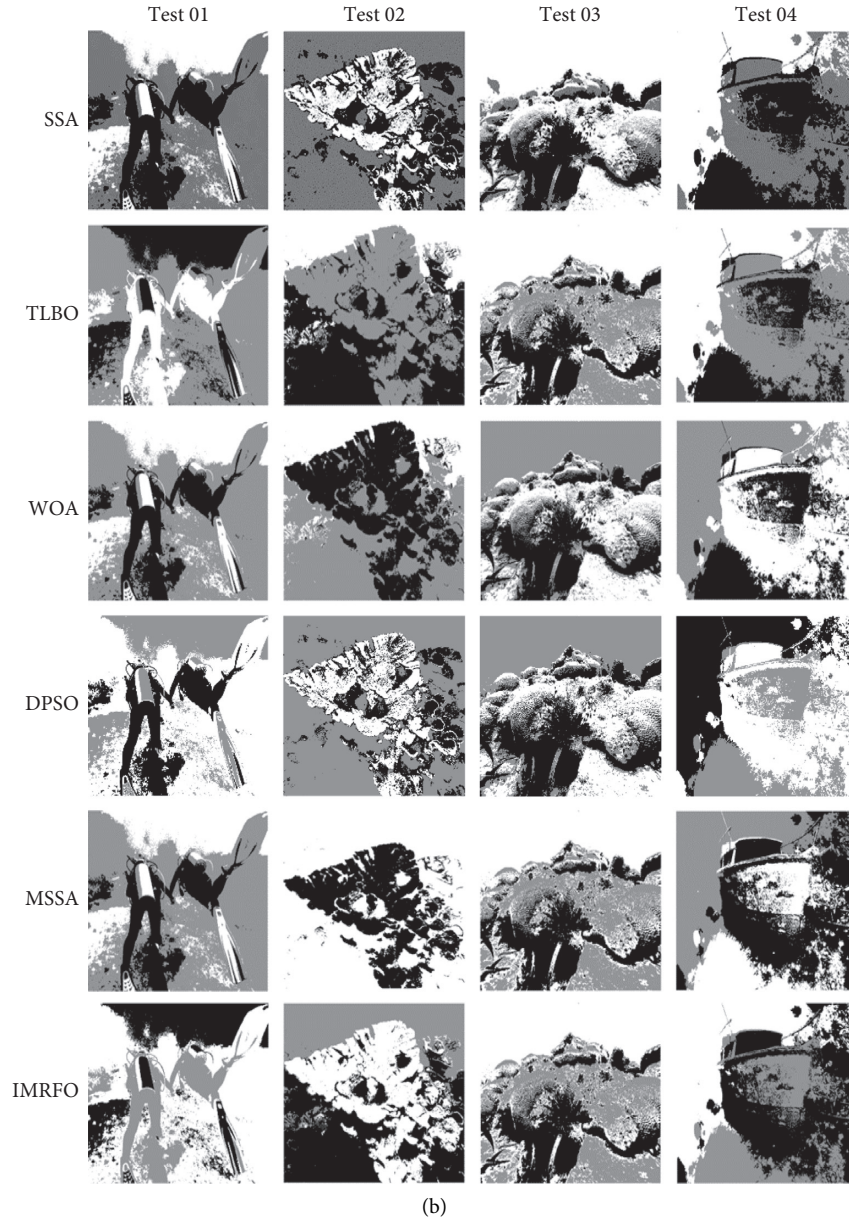
(a)

FIGURE 5: Continued.

(b)

FIGURE 5: Segmentation effect of test 01–4.

7. Summary of Results

MRFO relies on group behavior to find food, so it lacks flexibility and is prone to fall into local optimum. In the existing work, most scholars can not solve such problems well. In order to improve the searching ability of MRFO, an improved algorithm for bats is presented, which uses Lévy flight, random walk learning, and learning factors.

The current experimental work is summarized as follows:

(1) Comparing IMRFO with some basic algorithms on 12 standard test functions shows that the algorithm has certain advantages.

(2) Two statistical tests are used to verify the universality of IMRFO and show a good search ability.

(3) The convergence of each algorithm in each function is given, and the result shows that IMRFO has a good convergence rate.

(4) To further verify the performance of the algorithm, IMRFO is compared with the recently proposed variants of the algorithm, and the results show that IMRFO has an obvious advantage in most functions.

(5) In order to verify the validity and value of the three combinations of strategies, ablation experiments were carried out. The results show that IMRFO is better than other combinations of strategies, highlighting the practical value of IMRFO.

(6) Eight underwater images were used to verify the effect of IMRFO optimized K-means image
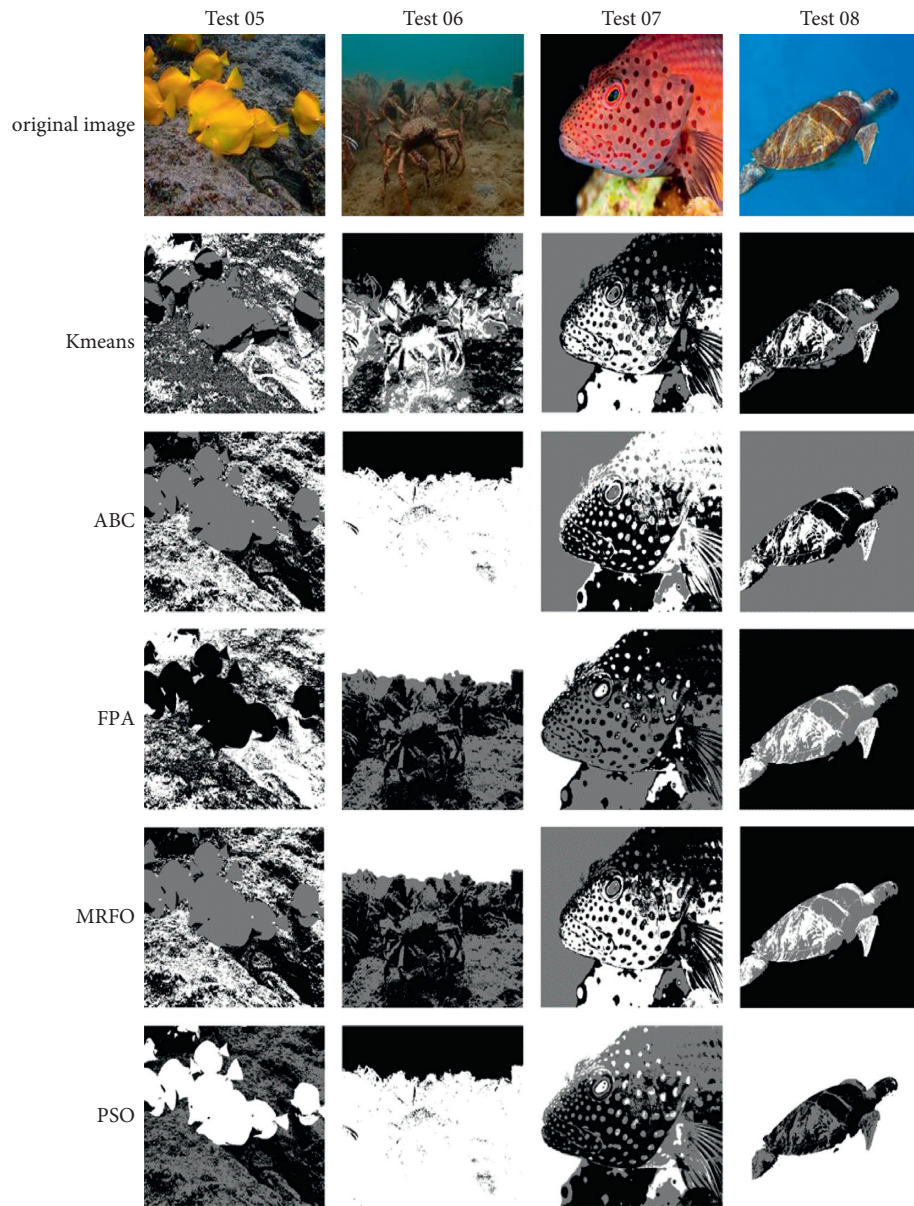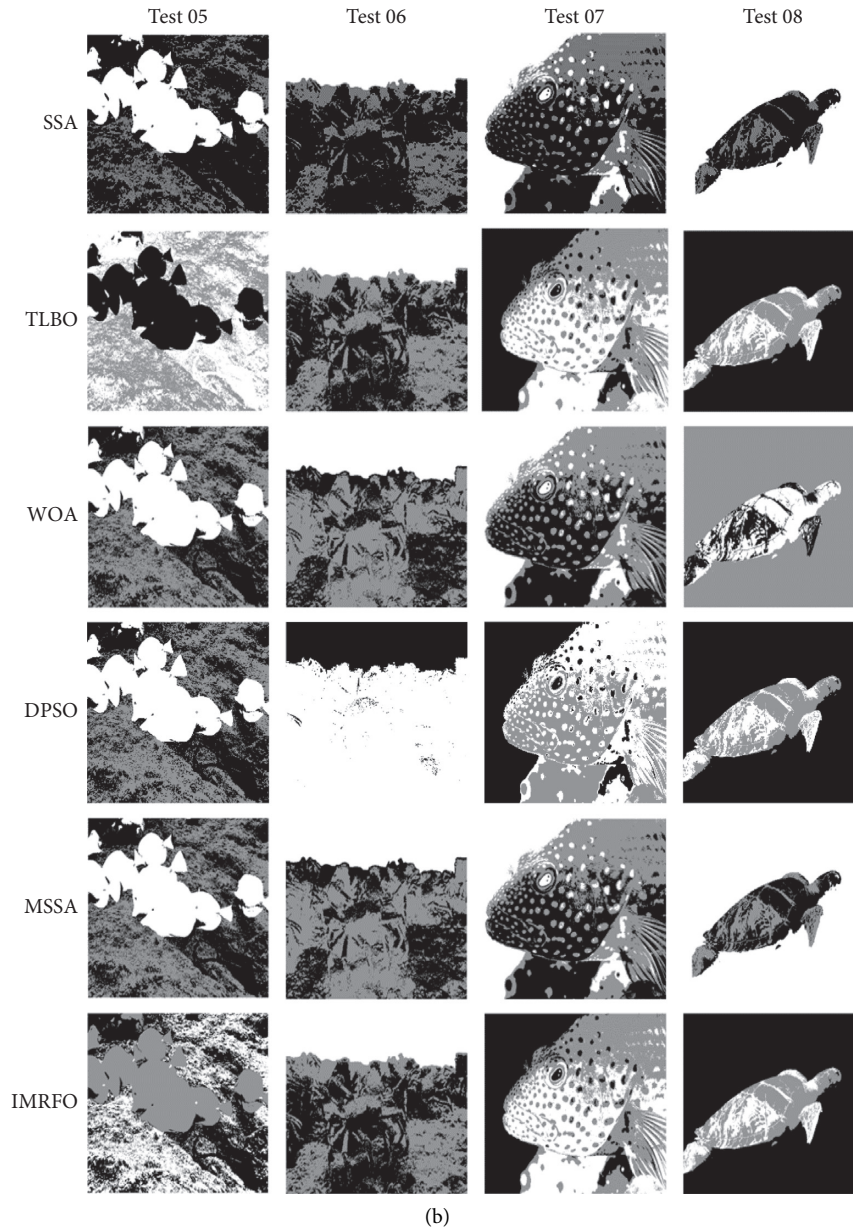
(a)

FIGURE 6: Continued.

(b)

Figure 6: Segmentation effect of test 05–8.

Table 8: Segmentation effect tables for each algorithm.

Image	Index	K-means	PSO	MRFO	DPSO	MSSA	IMRFO
Test 01	PSNR	8.0602	42.9051	42.9934	42.8040	43.4789	**44.4989**
	SSIM	0.0525	0.7354	0.7412	0.7370	0.7671	0.7479
	FSIM	0.5556	0.9478	0.9476	0.9571	0.9467	**0.9579**
	Ave time	**2.5886**	39.2606	76.7789	38.4986	50.6846	76.3475
Test 02	PSNR	6.4532	44.0008	43.3155	44.0595	43.9037	43.7004
	SSIM	0.0306	0.7975	0.7788	**0.8027**	0.7877	0.7970
	FSIM	0.4934	0.9464	0.9473	0.9410	0.9508	**0.9601**
	Ave time	**5.9745**	43.9744	78.42064	38.7315	50.7706	74.5214
Test 03	PSNR	6.6314	43.1602	43.3925	43.3760	43.2782	**43.7128**
	SSIM	0.0273	0.7615	0.7770	0.7655	0.7678	**0.7894**
	FSIM	0.4249	0.9400	0.9470	0.9424	**0.9506**	0.9430
	Ave time	**2.4941**	42.6951	75.7196	38.5346	50.7550	74.7570

TABLE 8: Continued.

Image	Index	K-means	PSO	MRFO	DPSO	MSSA	IMRFO
Test 04	PSNR	7.6085	43.0707	42.7419	43.2700	42.6999	43.1534
	SSIM	0.0461	0.7559	0.7396	0.7651	0.7380	**0.7808**
	FSIM	0.4920	0.9351	0.9435	0.9334	0.9397	**0.9466**
	Ave time	**5.8564**	41.2824	77.9430	38.4732	52.1376	75.2728
Test 05	PSNR	7.7941	43.1295	42.9913	43.2999	42.6999	**43.5453**
	SSIM	0.0187	0.7542	0.7503	0.7637	0.7380	**0.7726**
	FSIM	0.4011	0.9551	0.9566	0.9596	0.9397	**0.9634**
	Ave time	**2.2253**	37.4426	73.5829	37.6144	48.2604	73.7482
Test 06	PSNR	9.6892	42.9705	42.7052	43.539	42.8316	43.1626
	SSIM	0.0393	0.7375	0.7309	0.7385	0.7338	**0.7517**
	FSIM	0.5301	0.9501	0.9567	0.9480	0.9474	**0.9604**
	Ave time	**1.2299**	37.5475	72.6558	39.0477	48.7618	74.5432
Test 07	PSNR	7.82	42.7995	42.8791	72.8223	42.7924	**43.0107**
	SSIM	0.1071	0.7357	0.7399	0.7377	0.7414	0.7409
	FSIM	0.4460	0.9625	0.9584	0.9667	0.9558	**0.9734**
	Ave time	**2.3464**	39.1573	74.3283	38.1076	49.2879	74.5704
Test 08	PSNR	8.2165	44.1114	43.5191	44.5560	42.8239	**46.4141**
	SSIM	0.0210	0.7681	0.7417	0.7835	0.7185	**0.8710**
	FSIM	0.6614	0.8259	0.8130	0.8064	0.8444	0.8340
	Ave time	**1.2530**	38.8315	74.7158	38.2203	48.7720	75.4689
F	*index*	*WOA*	*ABC*	*TLBO*	*FPA*	*SSA*	*IMRFO*
Test 01	PSNR	43.1176	43.1018	42.8136	43.0370	43.4300	**44.4989**
	SSIM	**0.7488**	0.7473	0.7341	0.7407	0.7545	0.7479
	FSIM	0.9476	0.9441	0.9438	0.9441	0.9466	**0.9579**
	ave time	38.3036	39.4908	74.8217	40.1402	39.1653	76.3475
Test 02	PSNR	**44.1919**	43.5238	43.4408	42.9663	43.4401	43.7004
	SSIM	**0.8006**	0.7836	0.7852	0.7622	0.7808	0.7970
	FSIM	0.9482	0.9585	0.9557	0.9487	0.9531	**0.9601**
	ave time	38.8852	40.4172	75.3121	39.2922	42.0149	74.5214
Test 03	PSNR	43.4085	43.4408	43.0703	43.2635	43.3470	**43.7128**
	SSIM	0.7746	0.7790	0.7641	0.7731	0.7738	**0.7894**
	FSIM	0.9422	0.9453	0.9452	0.9442	0.9503	0.9430
	ave time	38.7937	42.8378	75.0803	41.8402	40.2512	74.7570
Test 04	PSNR	**43.3877**	43.2727	43.2877	42.7467	42.4343	43.1534
	SSIM	0.7600	0.7622	0.7630	0.7410	0.7256	**0.7808**
	FSIM	0.9426	0.9402	0.9403	0.9400	0.9380	**0.9466**
	ave time	39.2289	40.4403	74.4058	39.5984	39.0673	75.2728
Test 05	PSNR	42.9238	42.9262	42.9594	42.8405	42.9593	**43.5453**
	SSIM	0.7518	0.7514	0.7497	0.7480	0.7497	**0.7726**
	FSIM	0.9607	0.9605	0.9568	0.9586	0.9568	**0.9634**
	ave time	39.0783	38.9743	74.9810	38.6717	40.5572	73.7482
Test 06	PSNR	42.9003	**43.3715**	42.5365	42.5136	43.1584	43.1626
	SSIM	0.7356	0.7501	0.7107	0.7147	10.7503	**0.7517**
	FSIM	0.9396	0.9503	0.9444	0.9442	0.9511	**0.9604**
	ave time	37.5257	38.4848	76.3897	38.1784	37.8186	74.5432
Test 07	PSNR	42.8455	42.8353	42.8986	42.8399	42.8093	**43.0107**
	SSIM	0.7410	0.7386	0.7409	**0.7450**	0.7397	0.7409
	FSIM	0.9426	0.9551	0.9566	0.9367	0.9491	**0.9734**
	ave time	38.7532	39.6091	74.4956	38.6593	38.118674	74.5704
Test 08	PSNR	44.0623	45.5442	43.2560	45.3984	43.7115	**46.4141**
	SSIM	0.7663	0.8221	0.7315	0.8182	0.7545	**0.8710**
	FSIM	0.8381	0.8398	0.8196	0.8263	0.8316	0.8340
	ave time	39.3148	39.1180	76.6363	39.1871	38.8752	75.4689

The bold font here is to see the advantages and disadvantages of IMRFO. Although IMRFO has good performance, it has no advantage in time.

segmentation. The results show that the IMRFO optimized image segmentation quality is good and has a rational segmentation index in multiple images.

In summary, several experiments have demonstrated that IMRFO is a challenging new variant of the algorithm. IMRFO has shown good results in several test functions and

image segmentation, but the optimization results in some functions and images need to be improved. More work is waiting to improve its optimization capability.

8. Conclusion and Future Works

In order to improve the shortcomings of K-means image segmentation and its vulnerability to local optimization, this paper presents a K-means image segmentation method based on IMRFO. IMRFO uses Lévy flight to improve the individual searchability, proposes random walk learning to prevent the premature phenomenon of the algorithm, and finally uses learning factor to improve the convergence accuracy of the algorithm so as to improve the search of the algorithm. The validity and feasibility of IMRFO are verified by 12 test functions, and through 8 underwater image data sets, it can be seen that IMRFO has a good segmentation effect and is superior to other algorithms proposed in recent years under several indicators.

Although IMRFO has good segmentation advantages in eight images, it does not achieve the best three criteria for all images. From the experimental point of view, IMRFO is only the best in test 05, while the other test pictures are basically the two best. On the other hand, the running time of each algorithm is too large, and the accuracy is at the expense of time. In the future, we will improve the image quality from the following three aspects.

(1) Comprehensively improve the three performance indicators, making the three indicators the best

(2) Balance the time and the search ability of the algorithm to get the best performance in an effective time

(3) It can be used in agricultural, aerospace, medical, and other scenarios so that the algorithm can play a suitable role in different environments

Data Availability

Some data of this study are confidential, so the experimental data cannot be uploaded. These data can be obtained from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] E. Ardizzone, R. Pirrone, and O. Gambino, "Fuzzy c-means segmentation on brain mr slices corrupted by rf-inhomogeneity," in *Proceedings of the International Workshop on Fuzzy Logic and Applications*, Springer, Berlin, Heidelberg, 2007.

[2] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3781–3799, 2021.

[3] R. C. Hrosik, E. Tuba, E. Dolicanin, R. Jovanovic, and M. Tuba, "Brain image segmentation based on firefly algorithm combined with k-means clustering," *Studies in Informatics and Control*, vol. 28, pp. 167–176, 2019.

[4] H. Li, H. He, and Y. Wen, "Dynamic particle swarm optimization and K-means clustering algorithm for image segmentation," *Optik*, vol. 126, no. 24, pp. 4817–4822, 2015.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[6] S. Kapoor, I. Zeya, C. Singhal, and S. J. Nanda, "A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation," *Procedia Computer Science*, vol. 115, 2017.

[7] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[8] G.-G Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, IEEE, Bali, Indonesia, 2015.

[9] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[10] 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.

[11] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 5, IEEE, Orlando, FL, USA, October 1997.

[12] X. Xia, L. Gui, F. Yu et al., "Triple archives particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 50, pp. 4862–4875, 2019.

[13] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65–85, 1994.

[14] S.-C. Liu, Z.-G. Chen, Z.-H. Zhan, S.-W. Jeon, S. Kwong, and J. Zhang, "Many-objective job-shop scheduling: a multiple populations for multiple objectives-based genetic algorithm approach," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–15, 2021.

[15] S. Das and N. S. Ponnuthurai, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 4–31, 2010.

[16] Z. H. Zhan, Z. J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–15, 2019.

[17] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing*, IEEE, Coimbatore, India, 2010.

[18] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[19] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[20] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, Article ID 103300, 2020.

[21] A. Fathy, H. Rezk, and D. Yousri, "A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm," *Solar Energy*, vol. 207, pp. 305–316, 2020.

[22] E. H. Houssein, G. N. Zaki, A. A. Z. Diab, and E. M. Younis, "An efficient Manta Ray Foraging Optimization algorithm for parameter extraction of three-diode photovoltaic model," *Computers & Electrical Engineering*, vol. 94, Article ID 107304, 2021.

[23] E. H. Houssein, I. E. Ibrahim, N. Neggaz, M. Hassaballah, and Y. M. Wazery, "An efficient ECG arrhythmia classification method based on Manta ray foraging optimization," *Expert Systems with Applications*, vol. 181, Article ID 115131, 2021.

[24] M. Aly and H. Rezk, "A MPPT based on optimized FLC using manta ray foraging optimization algorithm for thermo-electric generation systems," *International Journal of Energy Research*, vol. 45, no. 9, pp. 13897–13910, 2021.

[25] M. G. Hemeida, A. A. Ibrahim, A.-A. A. Mohamed, S. Alkhalaf, and A. M. B. El-Dine, "Optimal allocation of distributed generators DG based Manta Ray Foraging Optimization algorithm (MRFO)," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 609–619, 2021.

[26] M. G. Hemeida, S. Alkhalaf, A.-A. A. Mohamed, A. A. Ibrahim, and T. Senjyu, "Distributed generators optimization based on multi-objective functions using manta rays foraging optimization algorithm (MRFO)," *Energies*, vol. 13, no. 15, p. 3847, 2020.

[27] A. M. Shaheen, A. R. Ginidi, R. A. El-Sehiemy, and S. S. M. Ghoneim, "Economic power and heat dispatch in cogeneration energy systems using manta ray foraging optimizer," *IEEE Access*, vol. 8, pp. 208281–208295, 2020.

[28] U. C. Ben, A. E. Akpan, C. C. Mbonu, and E. D. Ebong, "Novel methodology for interpretation of magnetic anomalies due to two-dimensional dipping dikes using the Manta Ray Foraging Optimization," *Journal of Applied Geophysics*, vol. 192, Article ID 104405, 2021.

[29] O. Akdag and C. Yeroglu, "Optimal directional overcurrent relay coordination using MRFO algorithm: a case study of adaptive protection of the distribution network of the Hatay province of Turkey," *Electric Power Systems Research*, vol. 192, Article ID 106998, 2021.

[30] K. K. Ghosh, R. Guha, S. K. Bera, N. Kumar, and R. Sarkar, "S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem," *Neural Computing & Applications*, vol. 33, no. 17, pp. 11027–11041, 2021.

[31] F. A Alturki, H. Mh Farh, A. A Al-Shamma'a, and K. AlSharabi, "Techno-economic optimization of small-scale hybrid energy systems using manta ray foraging optimizer," *Electronics*, vol. 9, p. 12, 2020.

[32] K. Elmaadawy, M. Abd Elaziz, A. H. Elsheikh, A. Moawad, B. Liu, and S. Lu, "Utilization of random vector functional link integrated with manta ray foraging optimization for effluent prediction of wastewater treatment plant," *Journal of Environmental Management*, vol. 298, Article ID 113520, 2021.

[33] M. Abd Elaziz, D. Yousri, M. A. Al-qaness, A. M. AbdelAty, A. G. Radwan, and A. A. Ewees, "A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation," *Engineering Applications of Artificial Intelligence*, vol. 98, Article ID 104105, 2021.

[34] M. H. Hassan, E. H. Houssein, M. A. Mahdy, and S. Kamel, "An improved manta ray foraging optimizer for cost-effective emission dispatch problems," *Engineering Applications of Artificial Intelligence*, vol. 100, Article ID 104155, 2021.

[35] H. Xu, H. Song, C. Xu, X. Wu, and N. Yousefi, "Exergy analysis and optimization of a HT-PEMFC using developed manta ray foraging optimization algorithm," *International Journal of Hydrogen Energy*, vol. 45, no. 55, pp. 30932–30941, 2020.

[36] E. H. Houssein, M. M. Emam, and A. A. Ali, "Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images," *Neural Computing & Applications*, vol. 33, no. 24, pp. 16899–16919, 2021.

[37] B. Jena, M. K. Naik, R. Panda, and A. Abraham, "Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization," *Engineering Applications of Artificial Intelligence*, vol. 103, Article ID 104293, 2021.

[38] M. Micev, M. Ćalasan, Z. M. Ali, H. M. Hasanien, and S. H. E. Abdel Aleem, "Optimal design of automatic voltage regulation controller using hybrid simulated annealing - manta ray foraging optimization algorithm," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 641–657, 2021.

[39] S. Ekinci, D. Izci, and B. Hekimoğlu, "Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm," *Arabian Journal for Science and Engineering*, vol. 46, no. 2, pp. 1395–1409, 2021.

[40] H. . Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[41] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with levy flight for global optimization," *Applied Soft Computing*, vol. 43, pp. 248–261, 2016.

[42] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with Levy flight," *IEEE Access*, vol. 8, pp. 67205–67213, 2020.

[43] X. Li, A. Engelbrecht, and M. G. Epitropakis, *Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization*, RMIT University, Evolutionary Computation and Machine Learning Group, Australia, 2013.

[44] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test Functions for Optimization Needs*, vol. 101, p. 48, 2005.

[45] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey Badger Algorithm: new meta-heuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022.

[46] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[47] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[48] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, 2012.

[49] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[50] H. Peng, Z. Zeng, C. Deng, and Z. Wu, "Multi-strategy serial cuckoo search algorithm for global optimization," *Knowledge-Based Systems*, vol. 214, Article ID 106729, 2021.

[51] H. Peng, W. Zhu, C. Deng, and Z. Wu, "Enhancing firefly algorithm with courtship learning," *Information Sciences*, vol. 543, pp. 18–42, 2021.

[52] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Self adaptive cuckoo search: analysis and experimentation," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100751, 2021.

[53] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Improving cuckoo search: incorporating changes for CEC 2017 and CEC 2020 benchmark problems," in *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2020.

[54] M. Islam, P. Luo, and J. Sattar, "Simultaneous enhancement and super-resolution of underwater imagery for improved visual perception," 2020, https://www.semanticscholar.org/paper/Simultaneous-Enhancement-and-Super-Resolution-of-Islam-Luo.

[55] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[56] G. Chen, L. Dong, C. Dong, and X. Chen, "Image segmentation based on logistic regression sparrow algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 1-14, 2021.

[57] 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, pp. 242–256, 2017.

[58] P. Upadhyay and J. K. Chhabra, "Kapur's entropy based optimal multilevel image segmentation using Crow Search Algorithm," *Applied Soft Computing*, vol. 97, Article ID 105522, Glasgow, UK, 2020.

[59] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019.