Multimedia Tools and Applications (2023) 82:21825–21863
https://doi.org/10.1007/s11042-022-14024-2

TRACK 3: BIOMETRICS AND HCI

Kapur's entropy underwater image segmentation based on multi-strategy Manta ray foraging optimization

Donglin Zhu[1] · **Changjun Zhou**[1] · **Yaxian Qiu**[2] · **Feng Tang**[2] · **Shaoqiang Yan**[3]

Received: 15 November 2021 / Revised: 8 September 2022 / Accepted: 23 September 2022 /
Published online: 10 October 2022

Abstract

Image segmentation is an important part of image processing, which directly affects the quality of image processing results. Threshold segmentation is the simplest and most widely used segmentation method. However, the best method to determine the threshold has always been a NP-hard problem. Therefore, this paper proposes Kapur's entropy image segmentation based on multi-strategy manta ray foraging optimization, which has a good effect in CEC 2017 test function and image segmentation. Manta ray foraging optimization (MRFO) is a new intelligent optimization algorithm, which has good searchability, but the local development ability is insufficient, so it can not effectively find a reliable point. To solve this defect, this paper proposes a multi-strategy learning manta ray foraging optimization algorithm, referred to as MSMRFO, which uses saltation learning to speed up the communication within the population and improve the convergence speed, and then puts forward a behavior selection strategy to judge the current situation of the population, Tent disturbance and Gaussian mutation are used to avoid the algorithm falling into local optimization and improve the convergence speed of the algorithm. In the complete CEC 2017 test set, MSMRFO is compared with 8 algorithms, including FA_CL and ASBSO are variants of new algorithms proposed in recent years. The results show that MSMRFO has good optimization ability and universality. In nine underwater image data sets, MSMRFO has better segmentation quality than the other eight algorithms, and the segmentation indicators under high threshold processing has better advantages.

✉ Donglin Zhu
6920190624@mail.jxust.edu.cn

✉ Changjun Zhou
zhou–chang231@163.com

[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 341000 Jiangxi, China

[3] Xi'an Research Institute of High Technology, Xi'an 710025 Shanxi, China

Keywords Image segmentation · Kapur's entropy · Manta ray foraging optimization · Saltation learning · Tent disturbance · Gaussian mutation · CEC 2017 · Underwater image

1 Introduction

With the rapid development of computer technology, computer vision has gradually refined and formed its scientific system, in which image segmentation, as an important branch of the field of image processing, plays an increasingly important role. Image segmentation refers to the division of images into disjoint, meaningful sub-regions, the pixels in the same area have a certain correlation, and the pixels in different areas have certain differences, that is, the process of assigning the same label to pixels with the same nature in the picture [18]. Threshold segmentation is one of the classical segmentation techniques, and it is also the simplest, most practical, and efficient method [42]. The target of threshold segmentation is to select one or more specific thresholds to divide the image into several different regions. The main purpose of the threshold segmentation method is to obtain the most appropriate and effective thresholds for image segmentation. Common segmentation methods are the maximum class method, minimum error method, maximum entropy method, and so on. Kapur's entropy is also a popular segmentation method in recent years and has been applied to many fields by many researchers. This method is an image segmentation technique based on the entropy threshold transformation, which combines the probability distribution of image histogram in the process of use. When the threshold value is selected accurately, the entropy will get the maximum value. Therefore, how to find the best threshold quickly and accurately is the focus of threshold segmentation. When threshold segmentation is performed by a common exhaustive search, the computation time is long and the limitations are large.

With the rise and development of swarm intelligence optimization algorithms, it has been widely used in image segmentation because of its strong global optimization ability and fast convergence speed, which has played a role in reducing the segmentation time and improving the accuracy of the segmentation. Akay et al. [3] applied classical particle swarm algorithm and artificial bee swarm algorithm to multi-threshold image segmentation. Kapur's entropy and Otsu were selected as fitness functions to search for thresholds to obtain the best segmentation results. Mohamed Abd El Aziz [1] uses WOA and Moth-Flame Optimization (MFO) to solve the multi-threshold image segmentation problem. The results show that the two algorithms have better segmentation quality than other algorithms. Fayad et al. [15] proposed an image segmentation algorithm based on ACO. Upadhyay [46] proposed the Crow search algorithm to handle the multi-threshold segmentation problem, which has a good segmentation effect. Xing [29] applies TLBO to image segmentation. Zhou Y et al. [58] proposed a moth swarm algorithm for image segmentation. At the same time, to further obtain the optimal segmentation effect, scholars improved some algorithms and better handle the threshold segmentation problem. For example, Wachs-Lopes G et al. [13] proposed an improved Firefly algorithm to deal with the threshold search problem. It uses Gaussian mutation and neighborhood strategy to improve search efficiency and global searchability. Wang [47] introduced Levy flight into the salp swarm algorithm, which showed better pioneering ability and searchability, and made the segmentation quality better. Yang Z et al. [54] proposed a non-revisiting quantum-behaved PSO (NRQPSO) algorithm for image segmentation. Xin Lv et al. [49] proposed a multi threshold segmentation method based on improved sparrow search algorithm (ISSA). ISSA adopts the idea of bird swarm algorithm to improve the search and development ability of the algorithm and can find the best threshold quickly and accurately. Bao x et al. [7] proposed a method to solve the multi-threshold segmentation problem based on differential Harris Hawks Optimization. The results show that HHO-DE is an effective color image segmentation tool. Jia et al. [28] proposed a mutation strategy, Harris Hawks Optimization, to handle the multi-threshold segmentation problem, and achieved good results in the quality of the segmentation. Zhao D et al. [56] proposed the horizontal and vertical search ACO, which effectively reduces the probability of the algorithm falling into the local optimal, has better searchability, and makes the segmentation result better. Ismail S G et al. [44] proposed a chaotic optimal foraging algorithm for leukocyte segmentation in microscopic images. Pare S et al. [36] proposed CS and egg-laying radius-cuckoo search optimizer to solve multilevel threshold problems for color images using different parametric analysis methods. However, the global optimization capabilities of the above algorithms are still inadequate and can fall into local states in complex datasets. Before optimization, a large number of experiments are needed to select appropriate algorithm parameters, which makes the workload and efficiency of these algorithms significantly unbalanced.

Manta ray foraging optimization (MRFO) is a new swarm intelligence optimization algorithm proposed in 2020. It is stronger than Particle Swarm Optimization (PSO) [31], Genetic Algorithm (GA) [48], Differential Evolution (DE) [9], Cuckoo Search (CS) [53], Gravitational Search Algorithm (GSA) [40], and Artificial Bee Colony (ABC) [30] in function optimization. It has the advantages of few parameters, easy to understand, and strong global optimization [57]. So far, it has been successfully applied to solar energy [14, 23], ECG [24], generator [6, 21], power system [20], cogeneration energy system [45], geophysical inversion problem [8], directional overcurrent relay [4], feature selection [17], hybrid energy system [5], and sewage treatment [12]. MRFO has flexible searchability and strong global searchability, but it lacks local development ability. For example, the ordered search between individuals will cause greater dependence and lack of initiative, resulting in a better overall search range, but poor local search performance.

Inspired by the above literature, this paper presents a multi-strategy learning manta ray foraging optimization (MSMRFO) algorithm. It introduces saltation learning, which enables individuals to communicate closely and obtain important information in different locations. Then, a behavior selection strategy is presented, which introduces Tent disturbance and Gauss mutation to prevent the convergence shortage and local optimum in the later stage. This strategy makes an important judgment on the current situation and effectively improves the global optimization ability of the algorithm. The specific workload and contributions of this article are as follows:

(1) Firstly, Saltation learning is introduced to speed up the information exchange of the population and improve the search efficiency of the algorithm.
(2) A behavioral selection strategy is designed, which uses Tent disturbance and Gauss mutation to improve Manta ray convergence and trap into local optimum.
(3) In the CEC 2017 test set, MSMRFO is compared with 8 algorithms, among which the firefly algorithm with courtship learning (FA_CL) [37] and ASBSO [55] proposed in recent years are also compared. The results verify that the algorithm has good searching ability and universality.
(4) MSMRFO is used to optimize the threshold segmentation. It is also the first time that MRFO performs threshold segmentation in the underwater image. Nine underwater image datasets are used to validate the optimal segmentation from different thresholds. The result shows that MSMRFO has better segmentation quality than other algorithms.

The main part of the paper is structured as follows: Section 2 mainly introduces the background knowledge, including Kapur's entropy and basic MRFO. Section 3 introduces and analyses the content and process of MSMRFO. Section 4 is to test and analyze the algorithms in CEC 2017. Section 5 describes the process of Kapur's entropy threshold segmentation based on MSMRFO. Section 6 introduces and analyses the threshold segmentation experiments of each algorithm. The seventh section summarizes the full text, and the last section discusses the future work.

2 Background

2.1 Multi-threshold segmentation based on Kapur entropy

Kapur's entropy is one of the early methods applied to single-threshold image segmentation, and it has been applied to the field of multi-threshold segmentation by many scholars. This segmentation method is a more effective image segmentation technique based on the entropy threshold transformation method, which combines the probability distribution of the image histogram. When the optimal threshold value is correctly selected and allocated, the maximum entropy will go. The ultimate goal of this method is to search for the optimal threshold value, which is the maximum direct value.

Assume that K is the gray level of 0-K-1 for a given picture, N is the total number of pixels, and $f(i)$ is the frequency of the i-th intensity level.

$$N = f(0) + f(1) + \cdots + f(K{-}1) \tag{1}$$

The probability of the i-th strength level can be expressed as:

$$p_i = f(i)/N \tag{2}$$

Assume there are G thresholds: $\{th_1, th_2, \cdots, th_G\}$, where $1 \le G \le K - 1$. Use these thresholds to divide a given picture into G + 1 classes, each represented by the following symbols:

$$\begin{array}{c} Class(0) = \{0, 1, 2, \cdots, th_1{-}1\} \\ Class(1) = \{th_1, th_1 + 1, \cdots, th_2{-}1\} \\ \vdots \\ Class(G+1) = \{th_{G-1}, th_{G-1} + 1, \cdots, th_G\} \end{array} \tag{3}$$

The combination entropy is obtained by calculating the sum of each type of entropy. Entropy-based methods are calculated as follows:

$$\begin{array}{c} E_0 = -\sum\limits_{i=0}^{i=th_1-1} \frac{p_i}{w_0}\ln\frac{p_i}{w_0}, w_0 = \sum\limits_{i=0}^{i=th_1-1} p_i \\ E_1 = -\sum_{i=th_1}^{i=th_2-1} \frac{p_i}{w_1}\ln\frac{p_i}{w_1}, w_1 = \sum_{i=th_1}^{i=th_2-1} p_i \\ \vdots \\ E_G = -\sum\limits_{i=th_G}^{i=K-1} \frac{p_i}{w_G}\ln\frac{p_i}{w_G}, w_G = \sum\limits_{i=th_G}^{i=K-1} p_\mathrm{i} \end{array} \tag{4}$$

Where E_i represents the entropy of class i. The final current function is as follows:

$$F(th) = E_0 + E_1 + \cdots + E_G \tag{5}$$

For the best threshold, the higher the $F(th)$ value, the better.

2.2 Manta ray foraging optimization

Inspired by the foraging behavior of manta rays, the algorithm is divided into three stages: chain foraging, spiral foraging, and somersault foraging.

2.2.1 Chain foraging

When manta rays are foraging, the higher the food concentration at a certain location, the better the location. Although the specific location of the best food source is not known, assuming that the location with the highest known food concentration is the best food source, manta rays will observe and swim to the best food source first. During swimming, the first manta ray moves to the best food source, while other manta rays move to the best food source and the manta ray in front of it at the same time, forming a foraging chain from head to tail. That is, in each iteration, each manta ray updates its position according to the best food source position found so far and the manta ray in front of it. The mathematical model of the chain foraging process can be expressed as follows:

$$x_i^d(t+1)=\begin{cases} x_i^d(t)+r\cdot\left(x_{best}^d(t)-x_i^d(t)\right)+\alpha\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=1\\ x_i^d(t)+r\cdot\left(x_{i-1}^d(t)-x_i^d(t)\right)+\alpha\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,3,\ldots,N\end{cases} \tag{6}$$

In eq. (6), $x_i^d(t)$ represents the d-dimensional information of the location of the i-th manta ray in the t-generation, and r is a random number subject to [0,1] uniform distribution. $\alpha=2\cdot r\cdot\sqrt{|\log(r)|}$ is the weight coefficient, $x_{best}^d(t)$ is the d-dimensional information of the best location found at present. The manta ray in position i depends on the manta ray in position i-1 and the best food location currently found. The update of the first manta ray depends on the optimal location.

2.2.2 Cyclone foraging

When a manta ray finds a high-quality food source in a certain space, each manta ray in the manta ray population will connect the head to the tail and spiral to the food source. During the aggregation process, the movement mode of the manta ray population changed from simple chain movement to spiral movement around the optimal food source. The cyclone foraging process can be represented by the following mathematical model:

$$(t+1)=\begin{cases} x_{best}^d(t)+r\cdot\left(x_{best}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=1\\ x_{best}^d(t)+r\cdot\left(x_{i-1}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,3,\ldots,N\end{cases} \tag{7}$$

Among them, $\beta=2e^{\frac{r_1(T-t+1)}{T}}\cdot\sin(2\pi r_1)$ represents the weight coefficient of helical motion, t is the maximum number of iterations, r_1 is the rotation factor and obeys the uniform random number of [0,1]. In addition, to improve the efficiency of group foraging, MRFO randomly generates a new location in the optimization process and then performs a spiral search at that location. Its mathematical model is:

$$x_i^d(t+1)=\begin{cases} x_{rand}^d(t)+r\cdot\left(x_{best}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=1\\ x_{rand}^d(t)+r\cdot\left(x_{i-1}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,3,\ldots,N\end{cases} \tag{8}$$

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

2.2.3 Somersault foraging

When a manta ray finds a food source, it regards the food source as a fulcrum, rotates around the fulcrum, and somersaults to a new position to attract the attention of other manta rays. For the manta ray population, somersault foraging is a random, local and frequent action, which can improve the foraging efficiency of the manta ray population. The mathematical model is as follows:

$$x_i^j(t+1) = x_i^j(t) + S\left(r_1 x_{best}^j(t) - r_2 x_i^j(t)\right) \;\; i = 1, \ldots, N \tag{9}$$

S is the somersault 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 bats somersault to locations in search space that are symmetrical to the optimal solution at their current location.

3 Manta ray foraging optimization based on fusion mutation and learning

3.1 Algorithm analysis

From these equations, it can be seen that more communication between individuals and orderly work can improve the searchability of the algorithm and perform a wide search. However, the lack of initiative among 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-bit complex problems, the change of the optimal position is similar, which results in less change in the two updates before and after the algorithm, and 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.

3.2 Related work

At present, Scholars are also constantly exploring new technologies to make MRFO play a better optimization ability. For example, Mohamed Abd Elaziz [2] will combine fractional calculus with MRFO to provide the direction of manta ray movement. CEC 2017 has verified the feasibility of the algorithm and applied it to image segmentation with good results. Mohamed H. Hassan [19] combines a gradient optimizer with MRFO to reduce the probability that the algorithm will fall into a local optimum and has been successfully applied in single- and multi-objective economic emission scheduling. Haitao Xu [51] uses adaptive weighting and chaos to improve MRFO to efficiently handle thermodynamic problems. Essam H. Houssein [25] uses reverse learning to initialize the population, enhances the diversity of the population, and applies it to threshold image segmentation problems with good segmentation quality. Bibekananda Jena [27] 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. Mihailo Micev [33] fuses SA with MRFO and applies it to the PID controller, which is better than other algorithms. In addition, Serdar Ekinci [11] uses a reverse learning and fusion simulated annealing algorithm to improve the convergence speed of the algorithm. It has good control performance when applied to the FOPID controller.

Although the above work has achieved some results, there are still some problems: Firstly, simple fusion can not show good results in different optimization environments. Secondly, adaptive strategy and reverse learning still have drawbacks in the face of high-dimensional complex problems, and can not jump out of local optimum later.

3.3 Proposed algorithm

3.3.1 Saltation learning (SL)

In the process of searching for the MRFO, the individuals are connected and the location update is only related to the optimal location, which results in a lack of learning ability and monotonous searching methods. Therefore, an individual learning behavior needs to be enhanced to improve the searchability of the algorithm in different environments.

Saltation learning is a new learning strategy proposed by Penghu et al. [38]. It can learn in different dimensions. It calculates candidate solutions through the best location, the worst location, and the randomly selected location, which increases the population diversity and has good searchability. This reduces the chance of falling into a local optimum. SL is described as follows:

$$x_{i,j}^{t+1} = x_{best,k}^{t} + r \cdot \left(x_{a,l}^{t} - x_{worst,n}^{t} \right) \tag{10}$$

In eq. (10), x_{best}^{t} and x_{worst}^{t} represents the best and worst position of the t iterations, *k, l, n* are three different integers selected from [1, D]. D is the dimension, r is the random number of [−1,1], exploring positions in different directions by changing the sign. *a* is a random integer of [1,P], and P is the population number. As shown in Fig. 1, assuming a dimension of 3, individuals from three different dimensions guide the selection of the next location, which accelerates information exchange within the population and improves search efficiency.

3.3.2 Gaussian mutation (GM)

A search chain is formed between individuals of the algorithm, which can perform a good search, but it is small on local development problems. Individuals are lazy and cannot search freely. Gauss mutation can solve this problem well and perform a good local search.

The Gauss variance comes from the Gauss distribution. Specifically, in the process of performing the variance, the original parameter value is replaced by a random number that fits the normal distribution of the mean μ and variance σ^2 [16, 22]. The variance equation is:

$$mutaion(x) = x(1 + N(0,1)) \tag{11}$$

In the eq. (11), x is the original parameter value, and N (0,1) indicates the expected value is 0. A random number with a standard deviation of 1; $mutaion(x)$ represents the value after the Gaussian mutation.

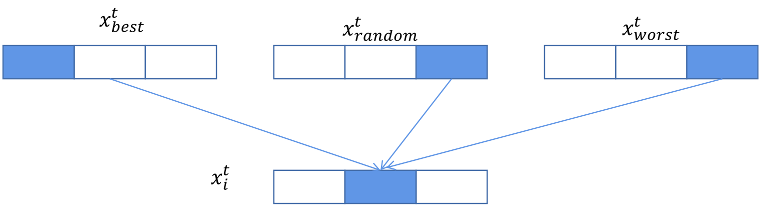


Fig. 1 SL Diagram

From the characteristics of normal distribution, it can be seen that the Gaussian distribution focuses on the local scope of the individual, carries out an efficient search, and improves the local search ability of the algorithm. For function problems with many local extremum points, it helps the algorithm to find global minimum points efficiently and accurately. it also improves the robustness of the algorithm.

3.3.3 Tent disturbance (TD)

Later individual manta rays are prone to fall into the local optimum, so chaotic disturbance is needed to make the algorithm jump out of the local optimum and improve the global searchability and optimization accuracy of the algorithm.

Chaos represents a nonlinear phenomenon in nature, so chaotic variables have the characteristics of randomness, ergodicity, and regularity, which can effectively improve the search efficiency of the algorithm. At present, the sequence generated by Tent mapping is more uniform than that generated by Logistic mapping. Therefore, using Tent mapping can effectively enable individuals to find quality positions [50]. The mathematical expression of Tent mapping is as follows:

$$x_{n+1} = \begin{cases} 2x_n, 0 \le x_n \le \frac{1}{2} \\ 2(1-x_n), \frac{1}{2} \le x_n \le 1 \end{cases} \tag{12}$$

The Tent mapping is expressed as follows after Bernoulli shift transformation:

$$x_{n+1} = (2x_n) mod\ 1 \tag{13}$$

Therefore, the steps of introducing Tent disturbance are as follows:

Step 1, generate chaotic variable x according to x_{n+1};

Step 2, apply chaotic variables to the solution of the problem to be solved:

$$X_d = min_d + (max_d - min_d) \cdot x_{n+1} \tag{14}$$

min_d and max_d are the minimum and maximum values of the d-th dimension x, respectively.

Step 3, make a chaotic disturbance to the individual according to the following equation:

$$newX' = (X' + newX)/2 \tag{15}$$

In the equation, X' represents the individual requiring chaotic perturbation, $newX$ is the chaotic variable generated, and $newX'$ is the individual after chaotic perturbation.

3.3.4 Selection of mutation and disturbance

First, to minimize the objective function, assume that F_{ave} is the average fitness value within the population. If the fitness value of an individual is less than F_{ave}, then clustering occurs. Gauss mutation makes these individuals slightly dispersed and improves the local search ability of the algorithm. Conversely, this means that individuals diverge, their current position is unreliable and disturbances are needed to improve their quality. Individuals after mutation and disturbance will change their position if they are better than those before, otherwise, the position will not change. The specific behavior selection (BC) equation is as follows:

$$x_i^d(t) = \begin{cases} GM, if\ F_i < F_{ave} \\ TD, if\ F_i \geq F_{ave} \end{cases} \quad (16)$$

$x_i^d(t)$ represents the updated individual, F_i represents the fitness value of the i-th individual.

3.3.5 Fusion multi-strategy learning manta ray foraging optimization

To improve the local development and learning ability of the manta ray foraging optimization, this paper proposes a multi-strategy learning manta ray foraging optimization algorithm. The algorithm uses saltation learning to speed up the internal communication of the population and improve the learning ability of the algorithm to adapt to different environments. Then a behavior selection strategy is presented, which uses Tent disturbance and Gauss mutation to balance the global search and local development capabilities of the algorithm by comparing the current and average fitness values, thus improving the quality of each optimal solution. The algorithm flow is as follows:
Algorithm: The framework of the MSMRFO.

```
Input
M: Maximum number of iterations
N: Population
Rand: Uniform random number of (0,1)
Output: X_best, f_g
Initialize population
t=1;
While(t<M)
Execute SL according to equation (10)
  For i=1 to N do
     If rand<0.5 then   // Cyclone foraging
       If t/M<rand then
       Update the position of the individual according to equation (8)
       Else
       Update the position of the individual according to equation (7)
       End if
     Else   // Chain foraging
    Update the position of the individual according to equation (6)
    End if
Compute the fitness of each individual, get X_best, f_g, X_worst
  For i=1 to N do   // Somersault foraging
  Update the position of the individual according to equation (9)
  End for
  Compute the fitness of each individual;
  The position of the individual is updated according to equation (16);
  Get a new X_best, f_g;
  End while
  Get the final X_best, f_g.
```

3.3.6 Time complexity analysis

Time complexity is an important index to measure an algorithm, so it is necessary to balance the optimization ability and time complexity of the algorithm in order to improve it effectively. The basic MRFO consists of only three stages: chain foraging, spiral foraging, and somersault foraging, in which chain foraging and spiral foraging are in the same cycle. Set the population number to N, the maximum number of iterations to T, and the dimension to D, so the time complexity of MRFO can be summarized as follows:

$$\begin{aligned} O(MRFO) &= O(T(O(cyclone\ foraging + chain\ foraging) + O(somersault\ foraging))) \\ &= O(T(ND + ND)) = O(TND) \end{aligned} \tag{17}$$

MSMRFO can be summarized as:

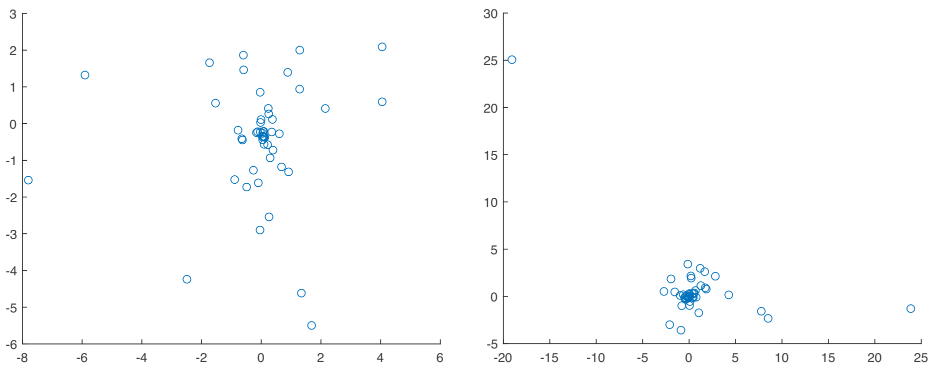
$$\begin{aligned} O(MSMRFO) = O(T(O(cyclone\ foraging + chain\ foraging) + O(SL) + O(somersault\ foraging) \\ +O(BC) + O(somersault\ foraging))) = O(TND) \end{aligned} \tag{18}$$

Therefore, it can be seen that the time complexity of MSMRFO has not changed radically.

3.3.7 Strategy effectiveness test

In order to test whether MSMRFO can really improve the optimization mechanism of the original algorithm, this paper takes sphere function as an example, and tests on MRFO and MSMRFO respectively. The population number is 50, the maximum number of iterations is 5, the theoretical optimal value of sphere function is 0, and the location is $x^* = (0, \ldots, 0)$. The final individual distribution of the two algorithms is given as shown in Fig. 2.

As shown in Fig. 2, it is clear that MRFO does not find an optimal value and is in a dispersed state, while MSMRFO has been clustered near the theoretical optimal value. Therefore, MSMRFO has a very fast convergence rate and a high accuracy. It can be seen that the introduction of multiple strategies significantly improves the optimization methods of the MRFO algorithm, speeds up the population exchange speed, and improves the quality of each solution obtained.



(a)MRFO (b)MSMRFO

Fig. 2 Algorithmic Personal Distribution Map (**a**)MRFO (**b**)MSMRFO

4 Performance testing

To verify the optimization capability of the MSMRFO algorithm, this paper tests each algorithm on the CEC 2017 test set, and compares eight algorithms with MSMRFO. The population number is 100, the number of assessments is 100 × D, and D is the dimension. To better reflect the effectiveness of the MSMRFO algorithm, this paper compares it with PSO, whale optimization algorithm (WOA) [34], sparrow search algorithm (SSA) [52], naked mole-rat algorithm (NMRA) [43], MRFO, Grey Wolf Optimizer (GWO) [35], and compares the two algorithms proposed in recent years, FACL and ASBSO, which have a good performance on the test set of CEC. PSO, GWO, and WOA are classical algorithms. SSA, MRFO, and NMRA are new swarm intelligence algorithms proposed in recent years. Because the parameters of some algorithms do not need special declaration and internal parameters do not need to be set, the parameters of some algorithms are shown in Table 1 In this paper, the Wilcoxon rank test is used to show whether there is a significant difference between each algorithm, which is tested at 5% significant level. "+" means that MSMRFO has more optimization capabilities than other algorithms, "-" conversely, "="means that the optimization performance between the two algorithms is equal, and "N/A" means that the values of the two algorithms are the same and cannot be compared. Specific test results are shown in Tables 2 and 3 in the Appendix.

The results of 30 operations of each algorithm are counted, and the five indexes of each algorithm, namely, optimal value, worst value, median, average value, and standard deviation, are calculated. In addition, the rank of each algorithm in each function is calculated, and the average rank is calculated to measure the universality of the algorithm. In order to clearly see the stability and optimization interval of each algorithm, a box diagram of 30 times the results of these functions in F3–6, F11–14, and F22–25 is given as shown in Fig. 3. These functions represent different types.

From Tables 2 and 3 and Fig. 3, we can see that MSMRFO has a great advantage in searchability and stability, especially in functions that show better searchability. Although some functions do not show better performance indicators, most of them have shown better search performance. Based on the fact that there is no free lunch theorem in the world, it is impossible to find an algorithm that performs well on any optimization problem, so MSMRFO is generally applicable. From the test results and average ranking, the average ranking of MSMRFO in the two tables is 1.34 and 1.7241 respectively. NMRA, PSO, and MRFO are second only to MSMRFO. And are the lowest ranking. MSMRFO has better advantages and is more perfect than the algorithm proposed in recent years. From the box diagram, the optimization effect of MSMRFO in each function is relatively stable, and the accuracy of the solution is also high. Generally speaking, the saltation learning and behavior selection strategy introduced by MSMRFO effectively avoids the phenomenon that the algorithm falls into local optimum and greatly improves the searchability of the algorithm.

Table 1 Parameters of each algorithm

Algorithm	SSA	GWO	PSO	ASBSO	FA-CL
Parameter	DS= $0.2\times N$ ST= $0.6\times$N	$a_{\max}=2$ $a_{\min}=0$	c1=c2=1.429 w=0.729	$K=5$	$\alpha=0.01$ $\beta_{min}=0.2$ $\beta=1$ $\gamma=1$

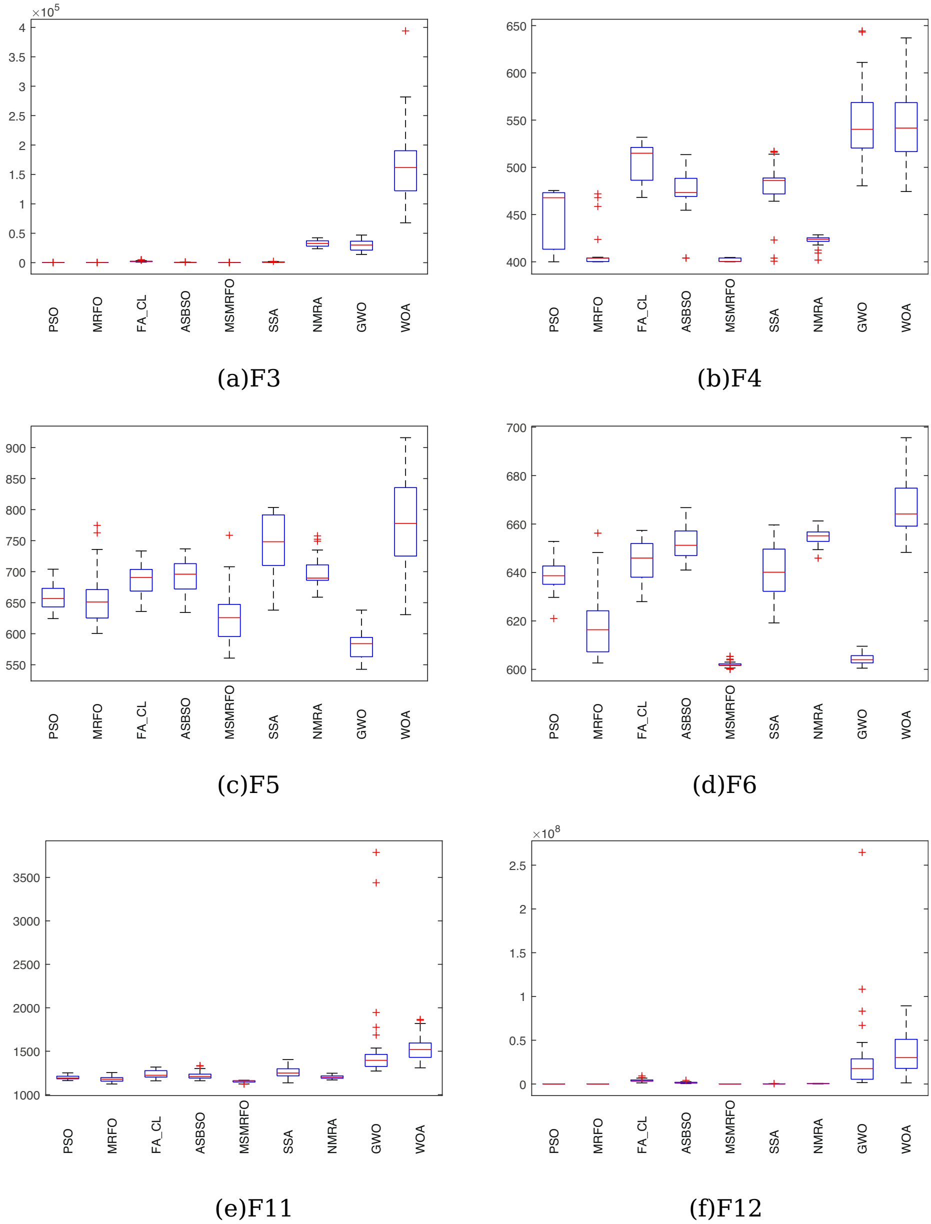
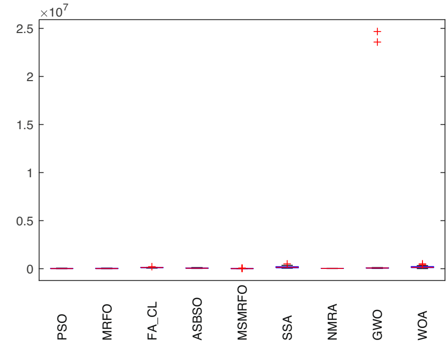


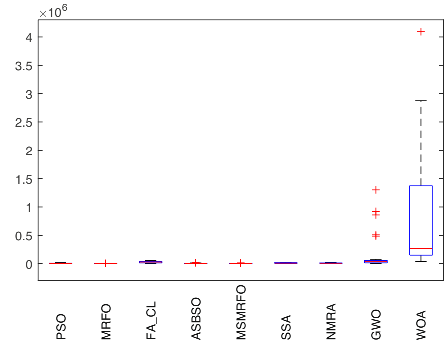
Fig. 3 Statistical chart of algorithm operation results (**a**)F3 (**b**)F4 (**c**)F5 (**d**)F6 (**e**)F11 (**f**)F12 (**g**)F13 (**h**)F14 (**i**)F22 (**j**)F23 (**k**)F24 (**l**)F25

5 Threshold segmentation process based on MSMRFO

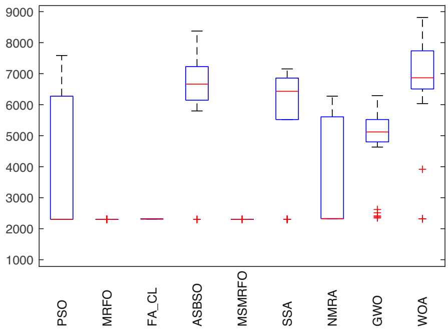
Assuming a k-dimensional threshold segmentation of the image, the solution vector is $T = [t_1, t_2, \cdots, t_k]$, which takes a positive integer and satisfies $0 < t_1 < t_2 < \cdots < t_n < L$. Multi-



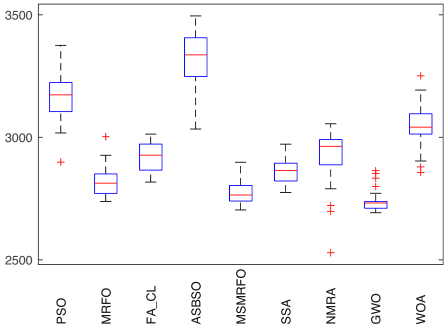
(g)F13



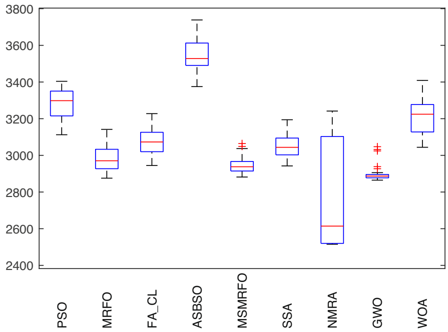
(h)F14



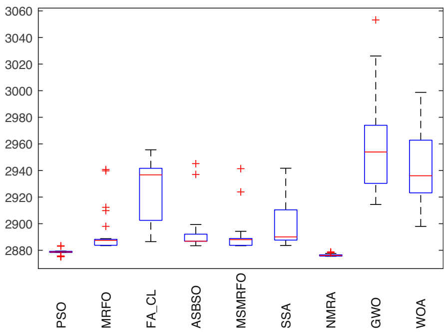
(i)F22



(j)F23



(k)F24



(l)F25

Fig. 3 (continued)

threshold segmentation is the process of finding a set of thresholds $[t_1, t_2, \cdots, t_k]$ (K > 0) in the image $f(x, y)$ to be segmented according to a certain criterion and dividing it into $K + 1$ parts. In this paper, Kapur's entropy is used as the segmentation criterion, MSMRFO is used to optimize the selection among L gray levels in solution space, and the maximization of eq. (1) is used as the objective function to solve. The multi-threshold segmentation process based on MSMRFO is shown in Fig. 4, and the detailed process is as follows:

Step1, Read the image to be split (grayscale image);
Step2, Get gray histogram of read-in image;
Step3, Initialization of MSMRFO parameters and setting of segmentation threshold K;
Step4, Initialization of the manta ray population. The individual position of a manta ray represents a threshold vector for image segmentation, and the component value of each vector ranges from [0,255] to an integer;
Step5, Perform MSMRFO;
Step6, if the algorithm reaches the preset end condition, the algorithm finishes the optimization and returns the best fitness of the bat location information. That is, the optimal threshold segmentation, otherwise jump to step 5.
Step7, Segment gray-scale images by the optimal threshold vector obtained, and output it.

6 Threshold segmentation experiment

6.1 Evaluating indicator

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

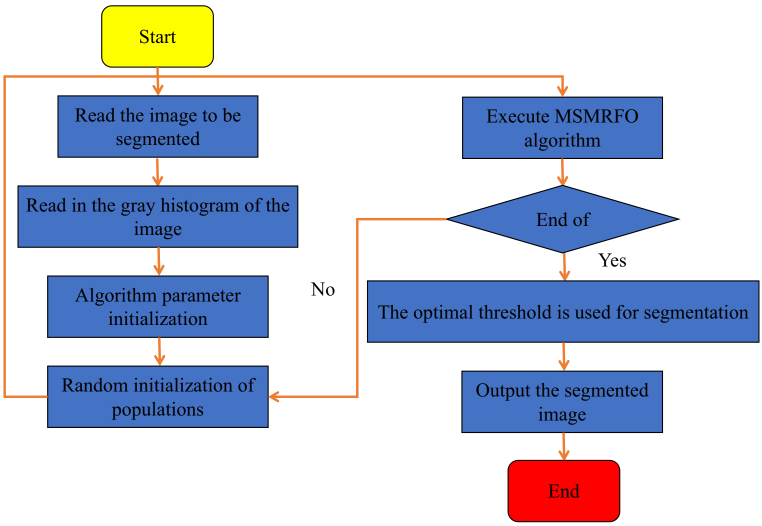


Fig. 4 MSMRFO-based threshold segmentation flowchart

PSNR is mainly used to measure the difference between the segmented image and the original image. The equation is as follows:

$$PSNR = 20 \cdot log_{10}\left(\frac{255}{RMSE}\right) \tag{19}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}(I(i,j)-Seg(i,j))^2}{M \times Q}} \tag{20}$$

In the equation, RMSE represents the root mean square error of the pixel; M × 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 image segmentation quality.

SSIM is used to measure the similarity between the original image and the segmented image. The larger the SSIM, the better the segmentation results. SSIM is defined as:

$$SSIM = \frac{\left(2\mu_I\mu_{seg} + c_1\right)\left(2\sigma_{I,seg} + c_2\right)}{\left(\mu_I^2 + \mu_{seg}^2 + c_1\right)\left(\sigma_I^2 + \sigma_{seg}^2 + c_2\right)} \tag{21}$$

In the equation, 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 between the original image and the segmented image; $\sigma_{I,\,seg}$ represents the covariance between the original image and the segmented image; c_1, c_2 are constants used to ensure stability.

FSIM is a measure of feature similarity between the original image and the segmentation quality, 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 result is. FSIM is defined as follows:

$$FSIM = \frac{\sum_{l\in\Omega}S_L(X)PC_m(X)}{\sum_{l\in\Omega}PC_m(X)} \tag{22}$$

$$SL(X) = \mathrm{S}_{PC}(X)S_G(X) \tag{23}$$

$$S_{PC}(X) = \frac{2PC_1(X)PC_2(X) + T_1}{PC_1^2(X)PC_2^2(X) + T_1} \tag{24}$$

$$S_G(X) = \frac{2G_1(X)G_2(X) + T_2}{G_1^2(X)G_2^2(X) + T_2} \tag{25}$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{26}$$

$$PC(X) = \frac{E(X)}{(\varepsilon + \sum_m A_n(X))} \tag{27}$$

In the above equation, Ω is all the pixel areas of the original image; $S_L(X)$ is the similarity score; $PC_m(X)$ is a measure of phase consistency; T_1 and T_2 are constant; G is a gradient descent; $E(X)$ is the size of the response vector at position X and the scale is n; ε is a very small number; $A_n(X)$ is the local size at scale n.

6.2 Experiment and analysis

To verify the effectiveness and feasibility of the MSMRFO algorithm, nine underwater image test sets [26] are selected in this paper. The underwater environment is complex and has a lot of debris, so the optimum performance of an algorithm can be tested most. At the same time, in order to prove that MSMRFO is competitive, it is compared with ten algorithms: MRFO, PSO, WOA, Teaching Learning Based Optimization (TLBO) [39], SSA, ISSA, GWO, BSA [32], CPSOGSA [41], HHO-DE. These ten algorithms have been applied to threshold image segmentation by researchers, so they are very persuasive. Each algorithm has experimented with 4 thresholds from 2 to 5. Each algorithm has a population of 30 and a maximum number of iterations of 100. A stop parameter of 10 is set in the experiment. If the solution found 10 times is the same in the optimization, it is assumed that the convergence has been completed. The significance of this is to reflect the value of the algorithm and find an algorithm with higher search efficiency. The experimental environment is Window10 64bit, the software is matlab2019b, the memory is 16GB, and the processor is Intel(R) Core(TM) i5-10200H CPU @ 2.40GHz. The MSMRFO algorithm segmentation image is shown in Fig. 5. The results of each algorithm are shown in Tables 4, 5, 6 and 7 in the Appendix. If MSMRFO has the best performance Indicators, the font will be bold. Among them, Table 4 shows the average fitness value ($F(th)$) of each algorithm to verify the optimization ability of the algorithm, and Tables 5, 6 and 7 show the PSNR, SSIM and FSIM performance indicators of each algorithm to verify the segmentation quality of the algorithm, respectively.

From Table 4, it can be seen that the number of optimal indicators of MSMRFO is higher relative to other algorithms, which shows the stronger optimization ability of MSMRFO, on the other hand, the indicators are not optimal in the case that its values are close to other optimal indicators want. Taken together, MSMRFO is effective in optimizing Kapur's Entropy and has sufficient generalizability.

It can see from Fig. 5, the image after MSMRFO segmentation becomes clearer with the increase of the threshold value, so MSMRFO has a good application value in threshold segmentation. From Table 5, 6 and 7, we can see that there are more optimal indicators for MSMRFO. In Test 05, the PSNR of each threshold segmentation is better than other algorithms. In Test 08, the SSIM indicators of each threshold segmentation is the best. In the individual images in Table 7, the FSIM indicators is optimal for MSMRFO with a threshold of 3 or more categories. Other algorithms have optimal criteria, but they are small in number and have better segmentation quality only at a certain threshold. Overall, MSMRFO has better segmentation quality at high thresholds and generally worse at low thresholds.

In order to better show the quality of MSMRFO segmentation under each threshold, the Friedman test [10] is applied to three performance indicators of each threshold, the ranking of each algorithm under different thresholds is calculated, and the final average rank is calculated to evaluate the segmentation effect of an algorithm. The test results are shown in Tables 8 and 9 in the Appendix. Table 8 shows the ranking results of MSMRFO and classical algorithms, and Table 9 shows the ranking of MSMRFO and the new algorithms and variant algorithms proposed in recent years. Similarly, if MSMRFO ranks best, its value will be bolded.

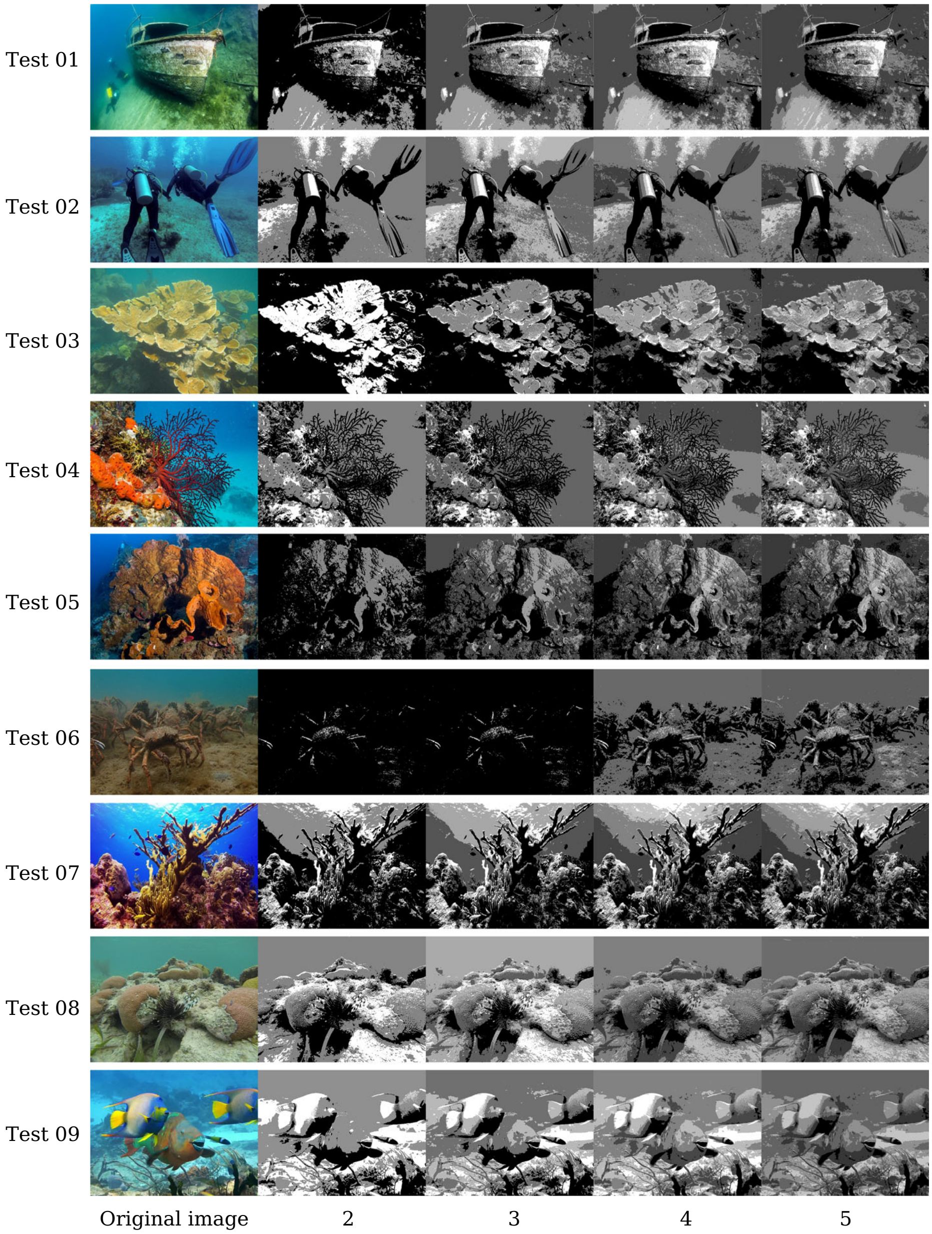


Fig. 5 Threshold segmentation image based on MSMRFO

It can be seen from Tables 8 and 9 that MSMRFO has a large number of optimal values, indicating that MSMRFO has a better optimization effect than the classical algorithm or compared with the algorithm proposed in recent years. At the same time, it also shows that MSMRFO has good universality in threshold segmentation and can show good application value in underwater images.

7 Conclusion

To better determine the optimal threshold in threshold segmentation, a Kapur's entropy image segmentation based on multi-strategy manta ray foraging optimization is presented. At the same time, a multi-strategy learning manta ray foraging optimization algorithm is proposed to improve the local development capability of the original algorithm and the probability of falling into the local optimum. The algorithm uses saltation learning to communicate among individuals, which accelerates the convergence speed of the algorithm. A new selection behavior strategy is proposed to make a better judgment on the current optimization stage, to prevent the algorithm from falling into local optimization and insufficient convergence, and to improve the global search ability of the algorithm. Tests on CEC 2017 show that the algorithm has good optimization ability and strong universality. Finally, nine underwater image data sets are segmented by MSMRFO. According to the segmented indicators, MSMRFO has better advantages and better quality in high threshold segmentation. From the Friedman test, MSMRFO ranked the highest, indicating that MSMRFO is generally good at segmentation in nine datasets.

8 Future work

Firstly, saltation learning has some randomness, so the final result is not maintained at a good level. Secondly, in some functions, no theoretical optimal value is found. At the same time, the fitness value does not yet achieve a certain advantage when optimizing the threshold. Thirdly, in many image data collections, it is not guaranteed that the three evaluation indicators in the same segmented image are optimal. For example, in Test 08, PSNR is not optimal with a threshold of 3, but the values of the other two indicators are the best. Finally, poor segmentation quality is common in low threshold processing. For example, in Test 02, none of the three MSMRFO indicators with a threshold of 3 was optimal. Therefore, The next step is to balance the optimization ability of the algorithm at high and low thresholds. In the aspect of image segmentation, we need to balance the quality of one image to ensure that the three indicators of image quality obtained at each time are excellent.

Appendix

Table 2 Test results of each algorithm in CEC2017

F	index	PSO [31]	MRFO [57]	FA_CL [37]	ASBSO [55]	MSMRFO
F1(X)	Best	1.00E+02	1.00E+02	8.21E+05	1.01E+02	1.00E+02
	Worst	1.21E+04	1.76E+04	1.85E+06	1.56E+05	1.12E+04
	Median	3.08E+03	1.44E+03	1.20E+06	2.08E+03	1.23E+03
	Mean	3.45E+03	3.78E+03	1.28E+06	8.26E+03	2.96E+03
	Std	3.27E+03	5.21E+03	2.96E+05	2.82E+04	3.66E+03
	P	4.73E-01(=)	7.73E-01(=)	3.02E-11(+)	3.02E-11(+)	
	rank	2	3	5	4	1
F3(X)	Best	3.00E+02	3.00E+02	8.98E+02	3.04E+02	3.00E+02
	Worst	3.00E+02	3.00E+02	5.10E+03	1.21E+03	3.00E+02
	Median	3.00E+02	3.00E+02	1.91E+03	3.97E+02	3.00E+02
	Mean	3.00E+02	3.00E+02	2.25E+03	5.01E+02	3.00E+02
	Std	4.62E-05	6.45E-07	2.17E+01	2.36E+02	0
	P	1.43E-08(+)	4.64E-03(+)	3.02E-11(+)	3.02E-11(+)	
	rank	3	2	5	4	1
F4(X)	Best	4.00E+02	4.00E+02	4.68E+02	4.04E+02	4.00E+02
	Worst	4.75E+02	4.72E+02	5.32E+02	5.13E+02	4.05E+02
	Median	4.68E+02	4.04E+02	5.15E+02	4.73E+02	4.00E+02
	Mean	4.45E+02	4.10E+02	5.06E+02	4.77E+02	4.02E+02
	Std	3.19E+01	1.97E+01	2.17E+01	2.60E+01	1.97E+00
	P	4.79E-07(+)	4.10E-01(+)	2.63E-11(+)	2.63E-11(+)	
	rank	3	2	5	4	1
F5(X)	Best	6.24E+02	6.00E+02	6.36E+02	6.34E+02	5.61E+02
	Worst	7.04E+02	7.75E+02	7.33E+02	7.37E+02	7.59E+02
	Median	6.57E+02	6.51E+02	6.91E+02	6.96E+02	6.26E+02
	Mean	6.58E+02	6.58E+02	6.84E+02	6.91E+02	6.32E+02
	Std	2.12E+01	4.37E+01	2.59E+01	2.93E+01	4.68E+01
	P	2.62E-03(+)	2.51E-02(+)	7.74E-06(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F6(X)	Best	6.21E+02	6.03E+02	6.28E+02	6.41E+02	6.00E+02
	Worst	6.53E+02	6.56E+02	6.57E+02	6.67E+02	6.05E+02
	Median	6.39E+02	6.16E+02	6.46E+02	6.51E+02	6.02E+02
	Mean	6.39E+02	6.19E+02	6.45E+02	6.52E+02	6.02E+02
	Std	7.10E+00	1.42E+01	9.34E+01	7.33E+00	1.31E+00
	P	2.72E-11(+)	1.96E-10(+)	2.72E-11(+)	2.72E-11(+)	
	rank	3	2	4	5	1
F7(X)	Best	8.35E+02	8.18E+02	8.86E+02	1.05E+03	7.71E+02
	Worst	1.02E+03	1.19E+03	1.26E+03	1.40E+03	8.96E+02
	Median	9.13E+02	9.64E+02	1.07E+03	1.17E+03	8.26E+02
	Mean	9.18E+02	9.70E+02	1.07E+03	1.19E+03	8.37E+02
	Std	4.15E+01	1.02E+02	9.34E+01	9.49E+01	2.47E+01
	P	1.00E-09(+)	7.34E-09(+)	3.42E-11(+)	2.80E-11(+)	
	rank	2	3	4	5	1
F8(X)	Best	8.66E+02	8.71E+02	8.74E+02	8.89E+02	8.65E+02
	Worst	9.72E+02	9.91E+02	9.80E+02	9.80E+02	9.79E+02
	Median	9.12E+02	9.37E+02	9.26E+02	9.28E+02	9.15E+02
	Mean	9.12E+02	9.32E+02	9.23E+02	9.32E+02	9.19E+02
	Std	2.38E+01	2.83E+01	2.42E+01	2.11E+01	2.56E+01
	P	8.68E-03(−)	7.17E-01(+)	2.77E-01(=)	3.02E-11(+)	
	rank	1	5	3	4	2
F9(X)	Best	2.65E+03	1.51E+03	2.27E+03	2.73E+03	9.07E+02
	Worst	4.58E+03	5.89E+03	6.12E+03	5.26E+03	3.17E+03

Table 2 (continued)

F	index	PSO [31]	MRFO [57]	FA_CL [37]	ASBSO [55]	MSMRFO
	Median	3.27E+03	3.14E+03	4.53E+03	3.95E+03	1.47E+03
	Mean	3.36E+03	3.17E+03	4.31E+03	3.91E+03	1.58E+03
	Std	5.22E+02	1.07E+03	6.15E+02	6.06E+02	5.73E+02
	P	3.10E-10(+)	2.79E-08(+)	1.18E-10(+)	2.95E-11(+)	
	rank	3	2	5	4	1
F10(X)	Best	3.30E+03	3.22E+03	3.61E+03	4.67E+03	2.64E+03
	Worst	6.11E+03	5.78E+03	6.19E+03	6.38E+03	5.76E+03
	Median	4.71E+03	4.44E+03	5.19E+03	5.32E+03	4.49E+03
	Mean	4.81E+03	4.54E+03	5.03E+03	5.42E+03	4.48E+03
	Std	6.95E+02	6.22E+02	6.15E+02	4.91E+02	7.65E+02
	P	1.09E-01(=)	8.77E-01(=)	4.86E-03(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F11(X)	Best	1.16E+03	1.12E+03	1.16E+03	1.16E+03	1.12E+03
	Worst	1.25E+03	1.25E+03	1.32E+03	1.33E+03	1.17E+03
	Median	1.19E+03	1.18E+03	1.22E+03	1.21E+03	1.16E+03
	Mean	1.20E+03	1.18E+03	1.23E+03	1.22E+03	1.15E+03
	Std	2.11E+01	3.44E+01	4.49E+01	4.37E+01	1.21E+01
	P	3.06E-11(+)	7.12E-04(+)	1.23E-10(+)	2.26E-11(+)	
	rank	3	2	5	4	1
F12(X)	Best	5.48E+03	7.64E+03	1.29E+06	4.76E+05	1.02E+04
	Worst	4.48E+04	8.34E+04	9.36E+06	4.25E+06	5.54E+04
	Median	2.02E+04	2.51E+04	3.59E+06	1.46E+06	3.61E+04
	Mean	2.13E+04	3.27E+04	4.18E+06	1.56E+06	3.27E+04
	Std	1.02E+04	2.01E+04	3.20E+04	7.72E+05	1.33E+04
	P	1.06E-03(−)	6.31E-01(=)	3.02E-11(+)	3.02E-11(+)	
	rank	1	2	5	4	3
F13(X)	Best	1.84E+03	1.36E+03	5.41E+04	1.62E+04	1.32E+03
	Worst	6.09E+04	6.23E+04	2.11E+05	1.17E+05	3.91E+04
	Median	1.10E+04	1.31E+04	1.19E+05	5.09E+04	7.12E+03
	Mean	1.80E+04	2.30E+04	1.13E+05	5.66E+04	1.11E+04
	Std	1.71E+04	2.21E+04	3.20E+04	2.86E+04	1.13E+04
	P	5.55E-02(=)	4.36E-02(+)	3.02E-11(+)	3.02E-11(+)	
	rank	2	3	5	4	1
F14(X)	Best	1.66E+03	1.50E+03	3.41E+03	1.79E+03	1.56E+03
	Worst	1.70E+04	5.80E+03	5.26E+04	1.98E+04	1.26E+04
	Median	3.52E+03	1.91E+03	2.86E+04	4.57E+03	2.59E+03
	Mean	6.33E+03	2.38E+03	2.54E+04	5.92E+03	3.73E+03
	Std	4.85E+03	1.06E+03	1.32E+04	4.41E+03	2.77E+03
	P	3.51E-02(+)	8.31E-03(−)	2.87E-10(+)	3.02E-11(+)	
	rank	3	1	5	4	2
F15(X)	Best	1.69E+03	1.52E+03	1.94E+04	1.21E+04	1.53E+03
	Worst	4.38E+04	2.88E+04	6.21E+04	8.21E+04	8.88E+03
	Median	5.70E+03	5.61E+03	4.36E+04	2.29E+04	1.80E+03
	Mean	9.54E+03	6.94E+03	4.08E+04	3.02E+04	2.71E+03
	Std	1.04E+04	6.19E+03	3.21E+02	1.90E+04	1.98E+03
	P	7.55E-06(+)	6.45E-04(+)	2.86E-11(+)	2.86E-11(+)	
	rank	3	2	5	4	1
F16(X)	Best	2.28E+03	1.86E+03	2.49E+03	2.61E+03	1.76E+03
	Worst	3.31E+03	3.10E+03	3.68E+03	3.82E+03	2.86E+03
	Median	2.73E+03	2.48E+03	3.07E+03	3.11E+03	2.31E+03
	Mean	2.74E+03	2.48E+03	3.09E+03	3.14E+03	2.29E+03
	Std	3.10E+02	2.94E+02	3.21E+02	3.24E+02	3.63E+02
	P	2.24E-05(+)	7.70E-02(=)	1.04E-09(+)	2.92E-11(+)	
	rank	3	2	4	5	1
F17(X)	Best	1.87E+03	1.80E+03	1.79E+03	1.99E+03	1.76E+03
	Worst	2.79E+03	2.53E+03	2.54E+03	3.08E+03	2.39E+03
	Median	2.16E+03	2.16E+03	2.08E+03	2.41E+03	2.07E+03

Table 2 (continued)

F	index	PSO [31]	MRFO [57]	FA_CL [37]	ASBSO [55]	MSMRFO
	Mean	2.23E+03	2.15E+03	2.13E+03	2.45E+03	2.07E+03
	Std	2.65E+02	2.03E+02	2.40E+02	2.62E+02	1.75E+02
	P	5.75E-02(=)	1.86E-01(=)	3.95E-01(=)	3.02E-11(+)	
	rank	4	3	2	5	1
F18(X)	Best	1.45E+04	9.99E+03	5.21E+04	5.26E+04	1.99E+04
	Worst	3.09E+05	1.80E+05	6.73E+05	3.76E+05	1.39E+05
	Median	8.43E+04	5.01E+04	2.36E+05	1.23E+05	6.96E+04
	Mean	9.43E+04	5.50E+04	2.47E+05	1.40E+05	6.95E+04
	Std	7.60E+04	3.39E+04	5.79E+05	7.53E+04	3.18E+04
	P	8.30E-01(=)	8.12E-04(−)	7.04E-07(+)	3.02E-11(+)	
	rank	3	1	5	4	2
F19(X)	Best	1.97E+03	2.11E+03	2.16E+05	3.60E+04	1.92E+03
	Worst	4.62E+04	4.93E+04	2.39E+06	2.52E+05	1.90E+04
	Median	6.58E+03	7.09E+03	1.39E+06	1.44E+05	3.15E+03
	Mean	1.05E+04	1.16E+04	1.32E+06	1.41E+05	5.26E+03
	Std	1.15E+04	1.04E+04	5.79E+05	5.28E+04	4.57E+03
	P	1.63E-02(+)	1.11E-03(+)	3.02E-11(+)	3.02E-11(+)	
	rank	2	3	5	4	1
F20(X)	Best	2.25E+03	2.19E+03	2.26E+03	2.38E+03	2.07E+03
	Worst	2.94E+03	2.80E+03	2.69E+03	3.06E+03	2.69E+03
	Median	2.59E+03	2.39E+03	2.31E+03	2.79E+03	2.33E+03
	Mean	2.60E+03	2.41E+03	2.37E+03	2.78E+03	2.36E+03
	Std	1.48E+02	1.30E+02	1.20E+02	1.85E+02	1.70E+02
	P	3.83E-06(+)	1.91E-01(=)	9.71E-01(=)	3.02E-11(+)	
	rank	4	3	2	5	1
F21(X)	Best	2.38E+03	2.37E+03	2.37E+03	2.43E+03	2.34E+03
	Worst	2.49E+03	2.50E+03	2.60E+03	2.57E+03	2.52E+03
	Median	2.43E+03	2.42E+03	2.45E+03	2.49E+03	2.40E+03
	Mean	2.43E+03	2.42E+03	2.45E+03	2.49E+03	2.41E+03
	Std	3.00E+01	3.17E+01	9.25E-01	3.67E+01	3.65E+01
	P	2.16E-03(+)	4.51E-02(+)	2.53E-04(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F22(X)	Best	2.30E+03	2.30E+03	2.31E+03	2.30E+03	2.30E+03
	Worst	7.58E+03	2.30E+03	2.31E+03	8.37E+03	2.30E+03
	Median	2.30E+03	2.30E+03	2.31E+03	6.66E+03	2.30E+03
	Mean	4.01E+03	2.30E+03	2.31E+03	6.45E+03	2.30E+03
	Std	2.17E+03	1.39E+00	9.25E-01	1.55E+03	1.06E+00
	P	1.47E-05(+)	3.41E-01(=)	4.11E-12(+)	4.11E-12(+)	
	rank	4	2	3	5	1
F23(X)	Best	2.90E+03	2.74E+03	2.82E+03	3.03E+03	2.69E+03
	Worst	3.38E+03	3.00E+03	3.01E+03	3.50E+03	2.87E+03
	Median	3.17E+03	2.81E+03	2.93E+03	3.34E+03	2.77E+03
	Mean	3.17E+03	2.82E+03	2.92E+03	3.32E+03	2.77E+03
	Std	1.04E+02	5.98E+01	5.79E+01	1.13E+02	4.67E+01
	P	3.02E-11(+)	7.70E-04(+)	1.96E-10(+)	3.02E-11(+)	
	rank	4	2	3	5	1
F24(X)	Best	3.11E+03	2.88E+03	2.94E+03	3.37E+03	2.88E+03
	Worst	3.40E+03	3.14E+03	3.23E+03	3.74E+03	3.06E+03
	Median	3.30E+03	2.97E+03	3.07E+03	3.53E+03	2.94E+03
	Mean	3.28E+03	2.98E+03	3.07E+03	3.54E+03	2.95E+03
	Std	9.40E+01	6.80E+01	2.19E+01	9.19E+01	4.68E+01
	P	3.02E-11(+)	4.51E-02(+)	1.43E-08(+)	3.02E-11(+)	
	rank	4	2	3	5	1
F25(X)	Best	2.88E+03	2.88E+03	2.89E+03	2.88E+03	2.88E+03
	Worst	2.88E+03	2.94E+03	2.96E+03	2.95E+03	2.91E+03
	Median	2.88E+03	2.89E+03	2.94E+03	2.89E+03	2.89E+03
	Mean	2.88E+03	2.89E+03	2.93E+03	2.89E+03	2.89E+03

Table 2 (continued)

F	index	PSO [31]	MRFO [57]	FA_CL [37]	ASBSO [55]	MSMRFO
	Std	1.93E+00	1.48E+01	2.19E+01	1.40E+01	6.07E+00
	P	3.02E-11(−)	4.20E-01(=)	1.07E-09(+)	3.02E-11(+)	
	rank	1	4	5	3	2
F26(X)	Best	2.80E+03	2.80E+03	2.83E+03	6.66E+03	2.80E+03
	Worst	9.27E+03	7.15E+03	7.86E+03	1.00E+04	6.64E+03
	Median	6.86E+03	5.18E+03	4.81E+03	8.61E+03	5.11E+03
	Mean	5.82E+03	4.86E+03	4.77E+03	8.57E+03	4.77E+03
	Std	2.21E+03	1.47E+03	1.87E+03	6.99E+02	1.28E+03
	P	1.18E-02(+)	8.47E-01(+)	8.42E-01(=)	2.92E-11(+)	
	rank	4	3	1	5	2
F27(X)	Best	3.16E+03	3.22E+03	3.36E+03	3.41E+03	3.21E+03
	Worst	3.90E+03	3.34E+03	3.67E+03	4.22E+03	3.34E+03
	Median	3.18E+03	3.26E+03	3.47E+03	3.81E+03	3.26E+03
	Mean	3.31E+03	3.27E+03	3.48E+03	3.81E+03	3.26E+03
	Std	2.34E+02	2.99E+01	9.39E+00	1.71E+02	2.60E+01
	P	1.95E-03(+)	8.19E-01(=)	3.02E-11(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F28(X)	Best	3.10E+03	3.10E+03	3.18E+03	3.10E+03	3.10E+03
	Worst	3.21E+03	3.27E+03	3.21E+03	3.25E+03	3.25E+03
	Median	3.10E+03	3.10E+03	3.20E+03	3.20E+03	3.10E+03
	Mean	3.12E+03	3.13E+03	3.20E+03	3.19E+03	3.12E+03
	Std	4.50E+01	5.67E+01	9.39E+00	4.15E+01	4.62E+01
	P	4.40E-01(=)	7.48E-01(=)	3.57E-07(+)	7.88E-12(+)	
	rank	3	2	5	4	1
F29(X)	Best	3.10E+03	3.33E+03	3.88E+03	3.89E+03	3.39E+03
	Worst	3.21E+03	4.34E+03	4.84E+03	4.99E+03	4.21E+03
	Median	3.10E+03	3.82E+03	4.34E+03	4.42E+03	3.76E+03
	Mean	3.12E+03	3.80E+03	4.33E+03	4.40E+03	3.79E+03
	Std	4.50E+01	2.66E+02	2.72E+02	2.85E+02	2.46E+02
	P	1.54E-01(=)	9.47E-01(=)	1.70E-08(+)	3.02E-11(+)	
	rank	1	3	4	5	2
F30(X)	Best	3.10E+03	5.53E+03	1.15E+06	9.21E+04	5.86E+03
	Worst	3.21E+03	1.40E+04	4.46E+06	1.45E+06	1.70E+04
	Median	3.10E+03	7.55E+03	2.21E+06	5.58E+05	7.64E+03
	Mean	3.12E+03	8.16E+03	2.52E+06	6.45E+05	8.55E+03
	Std	4.50E+01	2.31E+03	8.37E+05	3.53E+05	2.68E+03
	P	2.57E-07(−)	4.46E-01(−)	3.02E-11(+)	3.02E-11(+)	
	rank	1	2	5	4	3
	+/=/−	18/7/4	15/11/3	25/4/0	29/0/0	
	Average ranking	2.7241	2.3793	3.93	4.5862	1.34

Table 3 Test results of each algorithm in CEC2017

F	index	SSA [52]	NMRA [43]	GWO [35]	WOA [34]	MSMRFO
F1(x)	Best	1.00E+02	5.51E+04	8.91E+07	5.85E+05	1.00E+02
	Worst	2.05E+04	1.80E+05	2.28E+09	7.23E+06	1.12E+04
	Median	1.67E+03	1.02E+05	1.01E+09	1.90E+06	1.23E+03
	Mean	4.16E+03	1.05E+05	9.65E+08	2.43E+06	2.96E+03
	Std	5.78E+03	2.93E+04	6.71E+08	1.76E+06	3.66E+03
	P	5.49E-01(=)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	
	rank	2	3	5	4	1
F3(X)	Best	3.91E+02	2.37E+04	1.40E+04	6.78E+04	3.00E+02
	Worst	2.33E+03	4.23E+04	4.69E+04	3.94E+05	3.00E+02
	Median	7.93E+02	3.27E+04	3.01E+04	1.62E+05	3.00E+02
	Mean	9.66E+02	3.27E+04	2.92E+04	1.65E+05	3.00E+02
	Std	4.97E+02	4.90E+03	9.29E+03	6.73E+04	0
	P	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	
	rank	2	4	3	5	1
F4(X)	Best	4.01E+02	4.02E+02	4.80E+02	4.74E+02	4.00E+02
	Worst	5.17E+02	4.29E+02	6.44E+02	6.37E+02	4.05E+02
	Median	4.86E+02	4.24E+02	5.40E+02	5.42E+02	4.00E+02
	Mean	4.79E+02	4.22E+02	5.49E+02	5.45E+02	4.02E+02
	Std	2.83E+01	5.59E+00	4.02E+01	4.06E+01	1.97E+00
	P	2.30E-10(+)	1.06E-10(+)	2.63E-11(+)	2.63E-11(+)	
	rank	3	2	5	4	1
F5(X)	Best	6.38E+02	6.59E+02	5.43E+02	6.31E+02	5.61E+02
	Worst	8.03E+02	7.58E+02	6.38E+02	9.16E+02	7.59E+02
	Median	7.48E+02	6.90E+02	5.84E+02	7.78E+02	6.26E+02
	Mean	7.43E+02	6.98E+02	5.80E+02	7.83E+02	6.32E+02
	Std	4.77E+01	2.75E+01	2.14E+01	7.47E+01	4.68E+01
	P	4.18E-09(+)	3.81E-07(+)	2.88E-06(−)	1.07E-09(+)	
	rank	4	3	1	5	2
F6(X)	Best	6.19E+02	6.46E+02	6.00E+02	6.48E+02	6.00E+02
	Worst	6.60E+02	6.61E+02	6.10E+02	6.96E+02	6.05E+02
	Median	6.40E+02	6.55E+02	6.04E+02	6.64E+02	6.02E+02
	Mean	6.41E+02	6.55E+02	6.04E+02	6.67E+02	6.02E+02
	Std	1.08E+01	3.40E+00	2.09E+00	1.05E+01	1.31E+00
	P	2.72E-11(+)	2.72E-11(+)	4.75E-05(+)	2.72E-11(+)	
	rank	3	4	2	5	1
F7(X)	Best	9.60E+02	9.87E+02	7.92E+02	1.01E+03	7.71E+02
	Worst	1.34E+03	1.18E+03	9.80E+02	1.35E+03	8.96E+02
	Median	1.24E+03	1.11E+03	8.25E+02	1.22E+03	8.26E+02
	Mean	1.21E+03	1.11E+03	8.37E+02	1.20E+03	8.37E+02
	Std	1.10E+02	3.76E+01	4.64E+01	9.50E+01	2.47E+01
	P	2.80E-11(+)	2.80E-11(+)	1.41E-01(+)	2.80E-11(+)	
	rank	4	3	2	5	1
F8(X)	Best	9.07E+02	9.08E+02	8.47E+02	8.92E+02	8.65E+02
	Worst	1.03E+03	9.88E+02	9.78E+02	1.13E+03	9.79E+02
	Median	9.79E+02	9.42E+02	8.73E+02	9.90E+02	9.15E+02
	Mean	9.73E+02	9.44E+02	8.76E+02	9.91E+02	9.19E+02
	Std	3.06E+01	1.70E+01	2.35E+01	5.78E+01	2.56E+01
	P	7.74E-06(+)	3.03E-02(+)	2.03E-09(−)	8.29E-06(+)	
	rank	4	3	1	5	2
F9(X)	Best	3.43E+03	4.55E+03	9.66E+02	5.06E+03	9.07E+02
	Worst	5.58E+03	8.02E+03	2.33E+03	1.85E+04	3.17E+03
	Median	5.38E+03	6.31E+03	1.19E+03	8.53E+03	1.47E+03
	Mean	5.26E+03	6.28E+03	1.29E+03	9.10E+03	1.58E+03
	Std	3.68E+02	7.91E+02	3.14E+02	3.53E+03	5.73E+02
	P	2.95E-11(+)	2.95E-11(+)	1.83E-02(−)	2.95E-11(+)	
	rank	3	4	1	5	2

Table 3 (continued)

F	index	SSA [52]	NMRA [43]	GWO [35]	WOA [34]	MSMRFO
F10(X)	Best	3.49E+03	3.76E+03	2.86E+03	4.46E+03	2.64E+03
	Worst	6.12E+03	4.76E+03	7.36E+03	7.47E+03	5.76E+03
	Median	5.12E+03	4.39E+03	3.90E+03	6.19E+03	4.49E+03
	Mean	5.05E+03	4.38E+03	3.93E+03	6.13E+03	4.48E+03
	Std	6.17E+02	2.36E+02	8.08E+02	6.89E+02	7.65E+02
	P	4.43E-03(+)	6.00E-01(=)	7.70E-04(−)	2.67E-09(+)	
	rank	4	2	1	5	3
F11(X)	Best	1.14E+03	1.17E+03	1.27E+03	1.31E+03	1.12E+03
	Worst	1.40E+03	1.25E+03	3.79E+03	1.87E+03	1.17E+03
	Median	1.25E+03	1.20E+03	1.40E+03	1.52E+03	1.16E+03
	Mean	1.26E+03	1.20E+03	1.57E+03	1.53E+03	1.15E+03
	Std	6.99E+01	2.04E+01	5.78E+02	1.63E+02	1.21E+01
	P	9.60E-08(+)	2.26E-11(+)	2.26E-11(+)	2.26E-11(+)	
	rank	3	2	5	4	1
F12(X)	Best	4.73E+04	1.87E+05	1.61E+06	1.35E+06	1.02E+04
	Worst	4.96E+05	7.79E+05	2.65E+08	8.93E+07	5.54E+04
	Median	1.45E+05	4.26E+05	1.76E+07	3.02E+07	3.61E+04
	Mean	1.61E+05	4.75E+05	3.13E+07	3.52E+07	3.27E+04
	Std	9.61E+04	1.58E+05	5.05E+07	2.29E+07	1.33E+04
	P	7.39E-11(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	
	rank	2	3	4	5	1
F13(X)	Best	1.47E+03	1.31E+04	1.97E+04	9.26E+03	1.32E+03
	Worst	6.58E+04	4.53E+04	2.47E+07	5.09E+05	3.91E+04
	Median	1.26E+04	2.50E+04	6.89E+04	1.39E+05	7.12E+03
	Mean	1.86E+04	2.69E+04	1.67E+06	1.64E+05	1.11E+04
	Std	1.74E+04	9.26E+03	6.11E+06	1.07E+05	1.13E+04
	P	5.01E-02(+)	3.83E-06(+)	3.16E-10(+)	1.96E-10(+)	
	rank	2	3	5	4	1
F14(X)	Best	3.20E+03	2.93E+03	3.24E+03	3.54E+04	1.56E+03
	Worst	2.54E+04	2.03E+04	1.30E+06	4.09E+06	1.26E+04
	Median	9.98E+03	8.31E+03	4.21E+04	2.65E+05	2.59E+03
	Mean	1.12E+04	9.43E+03	1.65E+05	8.19E+05	3.73E+03
	Std	6.08E+03	4.62E+03	3.22E+05	1.05E+06	2.77E+03
	P	6.53E-08(+)	2.20E-07(+)	6.12E-10(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F15(X)	Best	1.77E+03	3.53E+03	7.50E+03	8.50E+03	1.53E+03
	Worst	4.18E+04	9.64E+03	3.65E+06	1.37E+05	8.88E+03
	Median	7.01E+03	6.60E+03	2.98E+04	5.53E+04	1.80E+03
	Mean	1.19E+04	6.61E+03	2.84E+05	6.39E+04	2.71E+03
	Std	1.18E+04	1.62E+03	8.14E+05	3.56E+04	1.98E+03
	P	7.38E-07(+)	4.91E-08(+)	3.50E-11(+)	3.17E-11(+)	
	rank	3	2	5	4	1
F16(X)	Best	2.31E+03	2.08E+03	1.88E+03	2.95E+03	1.76E+03
	Worst	3.54E+03	2.79E+03	3.17E+03	4.40E+03	2.86E+03
	Median	2.81E+03	2.52E+03	2.28E+03	3.40E+03	2.31E+03
	Mean	2.81E+03	2.49E+03	2.32E+03	3.47E+03	2.29E+03
	Std	3.19E+02	1.75E+02	2.66E+02	3.47E+02	3.63E+02
	P	1.70E-06(+)	4.04E-02(+)	7.50E-01(+)	2.92E-11(+)	
	rank	4	2	3	5	1
F17(X)	Best	1.96E+03	1.90E+03	1.79E+03	1.99E+03	1.76E+03
	Worst	2.97E+03	2.26E+03	2.42E+03	2.92E+03	2.39E+03
	Median	2.40E+03	2.04E+03	1.96E+03	2.49E+03	2.07E+03
	Mean	2.41E+03	2.05E+03	2.01E+03	2.47E+03	2.07E+03
	Std	2.44E+02	8.95E+01	1.51E+02	2.42E+02	1.75E+02
	P	4.80E-07(+)	5.11E-01(−)	1.49E-01(−)	1.25E-07(+)	
	rank	4	2	1	5	3
F18(X)	Best	2.16E+04	3.73E+04	5.43E+04	9.08E+04	1.99E+04

Table 3 (continued)

F	index	SSA [52]	NMRA [43]	GWO [35]	WOA [34]	MSMRFO
	Worst	4.35E+05	2.45E+05	6.16E+06	1.12E+07	1.39E+05
	Median	9.28E+04	1.07E+05	2.83E+05	1.83E+06	6.96E+04
	Mean	1.36E+05	1.06E+05	6.37E+05	2.61E+06	6.95E+04
	Std	1.06E+05	4.03E+04	1.12E+06	2.36E+06	3.18E+04
	P	1.15E-01(=)	1.33E-02(+)	1.10E-08(+)	2.61E-10(+)	
	rank	2	3	4	5	1
F19(X)	Best	2.05E+03	2.36E+03	4.05E+03	1.78E+05	1.92E+03
	Worst	5.00E+04	7.29E+03	1.44E+06	8.73E+06	1.90E+04
	Median	4.95E+03	4.70E+03	1.35E+05	1.99E+06	3.15E+03
	Mean	1.17E+04	4.68E+03	3.13E+05	2.91E+06	5.26E+03
	Std	1.42E+04	1.35E+03	4.42E+05	2.40E+06	4.57E+03
	P	3.51E-02(+)	8.50E-02(=)	1.33E-10(+)	3.02E-11(+)	
	rank	3	2	4	5	1
F20(X)	Best	2.13E+03	2.24E+03	2.17E+03	2.42E+03	2.07E+03
	Worst	3.03E+03	2.67E+03	2.60E+03	3.21E+03	2.69E+03
	Median	2.64E+03	2.46E+03	2.32E+03	2.82E+03	2.33E+03
	Mean	2.63E+03	2.45E+03	2.32E+03	2.82E+03	2.36E+03
	Std	2.16E+02	1.25E+02	8.77E+01	2.08E+02	1.70E+02
	P	5.46E-06(+)	3.03E-02(+)	4.20E-01(=)	1.69E-09(+)	
	rank	4	3	1	5	2
F21(X)	Best	2.39E+03	2.21E+03	2.34E+03	2.45E+03	2.34E+03
	Worst	2.61E+03	2.50E+03	2.46E+03	2.70E+03	2.52E+03
	Median	2.47E+03	2.43E+03	2.37E+03	2.58E+03	2.40E+03
	Mean	2.48E+03	2.35E+03	2.37E+03	2.58E+03	2.41E+03
	Std	5.61E+01	1.29E+02	2.49E+01	6.95E+01	3.65E+01
	P	2.15E-06(+)	9.71E-01(=)	3.83E-05(−)	9.92E-11(+)	
	rank	4	2	1	5	3
F22(X)	Best	2.30E+03	2.32E+03	2.34E+03	2.32E+03	2.30E+03
	Worst	7.15E+03	6.27E+03	6.29E+03	8.81E+03	2.30E+03
	Median	6.43E+03	2.33E+03	5.12E+03	6.87E+03	2.30E+03
	Mean	5.58E+03	3.41E+03	4.84E+03	6.65E+03	2.30E+03
	Std	1.88E+03	1.60E+03	1.16E+03	1.75E+03	1.06E+00
	P	3.24E-09(+)	4.11E-12(+)	4.11E-12(+)	4.11E-12(+)	
	rank	4	2	3	5	1
F23(X)	Best	2.77E+03	2.53E+03	2.69E+03	2.86E+03	2.69E+03
	Worst	2.97E+03	3.06E+03	2.86E+03	3.25E+03	2.87E+03
	Median	2.86E+03	2.96E+03	2.73E+03	3.04E+03	2.77E+03
	Mean	2.86E+03	2.92E+03	2.74E+03	3.05E+03	2.77E+03
	Std	5.76E+01	1.16E+02	4.27E+01	9.06E+01	4.67E+01
	P	6.53E-08(+)	3.01E-07(+)	8.68E-03(−)	3.69E-11(+)	
	rank	3	4	1	5	2
F24(X)	Best	2.94E+03	2.52E+03	2.86E+03	3.04E+03	2.88E+03
	Worst	3.19E+03	3.24E+03	3.05E+03	3.41E+03	3.06E+03
	Median	3.04E+03	2.61E+03	2.89E+03	3.22E+03	2.94E+03
	Mean	3.05E+03	2.75E+03	2.90E+03	3.21E+03	2.95E+03
	Std	7.07E+01	2.93E+02	4.76E+01	9.51E+01	4.68E+01
	P	8.35E-08(+)	7.96E-03(−)	2.88E-06(−)	3.69E-11(+)	
	rank	4	1	2	5	3
F25(X)	Best	2.88E+03	2.88E+03	2.91E+03	2.90E+03	2.88E+03
	Worst	2.94E+03	2.88E+03	3.05E+03	3.00E+03	2.91E+03
	Median	2.89E+03	2.88E+03	2.95E+03	2.94E+03	2.89E+03
	Mean	2.90E+03	2.88E+03	2.96E+03	2.94E+03	2.89E+03
	Std	1.72E+01	8.97E-01	3.32E+01	2.69E+01	6.07E+00
	P	2.25E-04(+)	3.02E-11(−)	3.02E-11(+)	4.50E-11(+)	
	rank	3	1	5	4	2
F26(X)	Best	2.80E+03	2.86E+03	4.00E+03	5.89E+03	2.80E+03
	Worst	7.58E+03	2.94E+03	4.83E+03	1.00E+04	6.64E+03

Table 3 (continued)

F	index	SSA [52]	NMRA [43]	GWO [35]	WOA [34]	MSMRFO
	Median	6.02E+03	2.90E+03	4.36E+03	7.96E+03	5.11E+03
	Mean	6.04E+03	2.90E+03	4.37E+03	7.85E+03	4.77E+03
	Std	8.26E+02	2.03E+01	1.98E+02	1.09E+03	1.28E+03
	P	2.32E-05(+)	8.74E-06(−)	8.26E-03(−)	1.06E-10(+)	
	rank	4	1	2	5	3
F27(X)	Best	3.22E+03	3.16E+03	3.21E+03	3.24E+03	3.21E+03
	Worst	3.34E+03	3.20E+03	3.26E+03	3.51E+03	3.34E+03
	Median	3.26E+03	3.19E+03	3.23E+03	3.33E+03	3.26E+03
	Mean	3.26E+03	3.19E+03	3.23E+03	3.35E+03	3.26E+03
	Std	3.07E+01	1.43E+01	1.45E+01	8.02E+01	2.60E+01
	P	6.73E-01(+)	3.02E-11(−)	6.28E-06(−)	6.53E-07(+)	
	rank	4	1	2	5	3
F28(X)	Best	3.10E+03	3.23E+03	3.27E+03	3.23E+03	3.10E+03
	Worst	3.26E+03	3.30E+03	3.42E+03	3.41E+03	3.25E+03
	Median	3.10E+03	3.29E+03	3.33E+03	3.31E+03	3.10E+03
	Mean	3.13E+03	3.29E+03	3.34E+03	3.31E+03	3.12E+03
	Std	5.89E+01	1.98E+01	3.54E+01	3.58E+01	4.62E+01
	P	1.39E-05(+)	1.20E-11(+)	7.88E-12(+)	1.08E-11(+)	
	rank	2	3	5	4	1
F29(X)	Best	3.61E+03	3.31E+03	3.43E+03	3.83E+03	3.39E+03
	Worst	4.84E+03	3.66E+03	3.93E+03	6.03E+03	4.21E+03
	Median	4.08E+03	3.49E+03	3.63E+03	4.86E+03	3.76E+03
	Mean	4.09E+03	3.50E+03	3.65E+03	4.85E+03	3.79E+03
	Std	2.89E+02	8.66E+01	1.60E+02	4.46E+02	2.46E+02
	P	2.53E-04(+)	2.00E-06(−)	2.92E-02(−)	1.96E-10(+)	
	rank	4	1	2	5	3
F30(X)	Best	5.38E+03	3.54E+03	7.45E+05	1.69E+06	5.86E+03
	Worst	2.05E+04	9.38E+03	1.22E+07	2.67E+07	1.70E+04
	Median	8.33E+03	5.39E+03	3.03E+06	1.07E+07	7.64E+03
	Mean	1.04E+04	5.64E+03	3.72E+06	1.16E+07	8.55E+03
	Std	4.04E+03	1.52E+03	2.92E+06	7.50E+06	2.68E+03
	P	6.57E-02(+)	8.20E-07(−)	3.02E-11(+)	3.02E-11(+)	
	rank	3	1	4	5	2
+/=/−		26/3/0	27/2/0	19/3/7	17/1/11	29/0/0
Average ranking		3.4213	2.3793	2.8966	4.7586	1.7241

Table 4 Average fitness values of each algorithm

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
Test 01	2	18.1101	17.9494	18.3995	18.4464	18.3691	**18.4485**
	3	22.6513	22.4544	23.3874	23.3871	23.3874	23.3841
	4	26.7138	26.5454	27.6898	27.6816	27.6905	**27.6954**
	5	30.2374	30.0912	31.6001	31.6047	31.5786	31.5846
Test 02	2	17.6492	17.4204	17.7055	17.7046	17.7055	**17.7056**
	3	21.8508	21.6544	21.9905	21.9904	21.9839	21.9909
	4	25.6895	25.4713	26.1013	26.1449	26.0824	26.1054
	5	29.3596	29.0889	30.2662	30.2629	30.0546	30.1439
Test 03	2	17.5255	17.5164	18.2076	18.1835	18.2077	**18.2077**
	3	21.9828	21.8330	22.9753	23.0088	23.0103	**23.0106**
	4	25.8697	25.8889	27.4243	27.3860	27.4262	**27.4288**
	5	29.4551	29.5242	31.5425	31.5478	31.5653	31.5651
Test 04	2	18.4149	18.1109	18.5691	18.5682	18.5691	18.5678
	3	22.6638	22.4539	23.2504	23.2475	23.1506	**23.2556**
	4	26.6622	26.5595	27.5072	27.5080	27.5094	27.5074
	5	30.3128	30.3428	31.4583	31.4495	31.4594	31.4595
Test 05	2	17.7205	17.6568	18.0427	18.0458	17.9774	18.0332
	3	22.3560	22.2336	22.8406	22.8352	22.8480	22.8337
	4	26.3894	26.2441	27.2503	27.2466	27.2519	**27.2607**
	5	30.1993	29.9985	31.3208	31.3072	31.3263	31.3016
Test 06	2	16.1128	16.0053	16.3351	16.3550	16.2957	16.3552
	3	20.2269	20.0573	20.8345	20.8941	20.8422	20.8473
	4	24.1351	23.9378	25.2907	25.3052	24.9314	25.3862
	5	27.6020	27.3745	29.2108	29.2551	28.6900	**29.5505**
Test 07	2	18.6188	18.4072	18.8369	18.8368	18.8370	**18.8370**
	3	23.1494	22.8151	23.4743	23.4740	23.4744	**23.4745**
	4	27.0966	26.7883	27.8066	27.8057	27.8072	27.8059
	5	30.7232	30.4228	31.8692	31.8659	31.8700	31.8620
Test 08	2	17.3229	17.2089	17.4145	17.4144	17.4145	**17.4145**
	3	21.7001	21.5035	21.8792	21.8793	21.8778	**21.8803**
	4	25.4711	25.4801	26.1594	26.1480	26.1613	**26.1738**
	5	29.3334	29.0700	30.2529	30.1606	30.1912	30.2286
Test 09	2	17.9269	17.6498	17.9952	17.9951	17.9953	**17.9953**
	3	22.0809	21.9220	22.5624	22.5578	22.5506	**22.5626**
	4	25.9091	25.7722	26.6971	26.6728	26.6772	**26.6977**
	5	29.3907	29.1290	30.5459	30.5006	30.5242	30.5407
F	index	MRFO [57]	SSA [52]	ISSA [49]	HHO-DE [7]	CPSOGSA [41]	MSMRFO
Test 01	2	18.4485	18.4516	18.4516	18.4468	18.4184	**18.4485**
	3	23.3842	23.3875	23.3874	23.3873	23.3875	23.3841
	4	27.6869	27.6903	27.6904	27.6871	27.6906	**27.6954**
	5	31.5932	31.6174	31.5945	31.5954	31.6013	31.5846
Test 02	2	17.7056	17.7056	17.7056	17.7055	17.7056	**17.7056**
	3	21.9894	21.9775	21.9880	21.9975	21.9838	21.9909
	4	26.1053	26.1301	26.1492	26.1811	26.1462	26.1054
	5	30.1369	30.2911	30.3708	30.2575	30.3799	30.1439
Test 03	2	18.2077	18.2077	18.1858	18.1858	18.2077	**18.2077**
	3	23.0104	23.0103	23.0104	22.9992	22.9757	**23.0106**
	4	27.4288	27.3964	27.4286	27.3903	27.3645	**27.4288**
	5	31.5675	31.4759	31.5456	31.4514	31.4550	31.5651
Test 04	2	18.5679	18.5691	18.5691	18.5690	18.5691	18.5678
	3	23.2527	23.2555	23.2555	23.2530	23.2540	**23.2556**
	4	27.5058	27.5071	27.5031	27.4887	27.4970	27.5074
	5	31.4583	31.4615	31.4575	31.4470	31.4555	31.4595
Test 05	2	18.0332	18.0476	18.0476	18.0458	18.0476	18.0332
	3	22.8337	22.8481	22.8481	22.8379	22.8480	22.8337
	4	27.2314	27.2530	27.2526	27.2473	27.2535	**27.2607**

Table 4 (continued)

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
	5	31.3034	31.3264	31.3249	31.3103	31.3254	31.3016
Test 06	2	16.3542	16.3151	16.3366	16.3773	16.2707	16.3552
	3	20.8486	20.8442	20.8504	20.9515	20.8778	20.8473
	4	25.2327	25.3353	25.5625	25.3660	25.6049	25.3862
	5	29.5035	29.4032	29.4851	29.2299	29.5019	**29.5505**
Test 07	2	18.8370	18.8370	18.8370	18.8370	18.8370	**18.8370**
	3	23.4745	23.4743	23.4743	23.4743	23.4745	**23.4745**
	4	27.8063	27.8072	27.8072	27.8063	27.8074	27.8059
	5	31.8663	31.8702	31.8679	31.8683	31.8713	31.8620
Test 08	2	17.4145	17.4145	17.4145	17.4145	17.4145	**17.4145**
	3	21.8803	21.8801	21.8754	21.8802	21.8803	**21.8803**
	4	26.1628	26.1691	26.1696	26.1479	26.1727	**26.1738**
	5	30.2410	30.2413	30.2063	30.2037	30.1660	30.2286
Test 09	2	17.9953	17.9953	17.9953	17.9952	17.9953	**17.9953**
	3	22.5626	22.5626	22.5626	22.5614	22.5625	**22.5626**
	4	26.6973	26.6541	26.6791	26.6420	26.6975	**26.6977**
	5	30.5539	30.5439	30.5699	30.5347	30.5644	30.5407

Table 5 PSNR segmentation effect table for each algorithm

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
Test 01	2	14.1803	14.0587	12.3746	11.2399	12.7880	11.1263
	3	16.7337	15.7164	17.2737	17.3199	17.2774	**17.2753**
	4	18.3546	18.2210	18.6832	18.6859	18.6832	18.6793
	5	19.8822	18.1557	20.1518	19.4918	21.1081	**21.5110**
Test 02	2	16.0865	15.4078	16.3965	16.4054	16.3979	**16.3979**
	3	18.3544	17.9259	18.7061	18.6257	19.0349	18.8173
	4	19.5056	18.6846	19.8795	18.9183	20.3492	**20.3948**
	5	20.7001	19.8701	20.2189	19.1859	21.5220	**21.5763**
Test 03	2	12.8387	12.0857	13.7890	13.6424	13.7890	**13.7890**
	3	14.1635	14.0040	14.7236	14.6674	14.7294	14.7347
	4	16.4136	16.4447	19.2432	18.3385	19.2584	**19.2777**
	5	17.7218	17.4951	20.5206	20.4337	20.6286	**20.6549**
Test 04	2	15.3100	14.0352	15.8161	15.8127	15.8169	15.8172
	3	17.2739	14.9589	16.1557	16.1441	16.8639	16.1971
	4	18.8683	17.9608	18.6589	18.6297	18.6485	18.6733
	5	19.0889	19.6792	21.0923	20.7651	20.9791	**21.7326**
Test 05	2	15.8803	14.4865	12.8845	12.8089	13.7240	12.8459
	3	17.1297	16.5005	17.7448	17.4570	17.7475	17.7525
	4	19.1481	17.2081	18.9640	18.8785	19.0056	**19.2697**
	5	20.6216	19.4558	19.9697	19.9815	20.3037	**20.6769**
Test 06	2	13.2103	13.9442	10.6537	9.9618	11.7544	10.0089
	3	15.9608	15.9494	16.1631	14.7534	16.8000	**16.8165**
	4	18.7178	18.0346	17.2448	16.3797	17.9103	17.0892
	5	19.4732	19.0989	19.3032	17.7794	21.2510	**21.3203**
Test 07	2	14.8035	14.2125	13.9569	13.9623	13.9540	13.9540
	3	17.0767	16.4539	17.6745	17.6768	17.6726	17.6743
	4	18.7913	17.6074	18.8777	18.8693	18.8842	**18.9149**
	5	18.9929	18.5958	19.7324	19.7559	19.7156	**19.9672**
Test 08	2	15.2174	14.9128	15.3116	15.2992	15.3116	**15.3116**
	3	17.5928	17.1629	19.1314	19.1320	19.0193	19.1258
	4	19.1003	17.9907	19.3640	19.1296	19.5244	**19.6154**
	5	19.6034	18.9588	19.6539	19.7929	21.0066	20.6399
Test 09	2	15.8102	14.9023	16.0311	16.0298	16.0316	16.0316
	3	17.6721	16.3706	17.1911	17.1692	17.2385	17.1961
	4	18.9335	18.2247	20.0008	19.8943	19.9113	**20.1072**
	5	19.4005	19.2759	20.7092	20.0857	20.9816	**21.0635**
F	index	MRFO [57]	SSA [52]	ISSA [49]	HHO-DE [7]	CPSOGSA [41]	MSMRFO
Test 01	2	11.1263	11.1263	11.1263	11.2311	11.9196	11.1263
	3	17.2748	17.2748	17.2751	17.2701	17.2748	**17.2753**
	4	18.6795	18.6718	18.6792	18.6710	18.6759	18.6793
	5	20.3771	19.6039	20.5458	20.2242	20.2492	**21.5110**
Test 02	2	16.3979	16.3979	16.3979	16.3993	16.3979	**16.3979**
	3	18.7839	19.3045	18.8261	18.0234	18.7486	18.8173
	4	19.9755	19.6042	19.4820	18.6514	19.6419	**20.3948**
	5	20.9798	20.5485	20.2183	19.4613	19.8442	**21.5763**
Test 03	2	13.7890	13.7890	13.6751	13.6751	13.7890	**13.7890**
	3	14.7318	14.7334	14.7318	14.7286	14.7441	14.7347
	4	19.2777	19.1231	19.2765	18.4135	18.9737	**19.2777**
	5	20.5847	20.3922	20.5260	20.0765	20.3175	**20.6549**
Test 04	2	15.8177	15.8169	15.8169	15.8161	15.8169	15.8172
	3	16.1957	16.1961	16.1956	16.1757	16.1800	16.1971
	4	18.6692	18.6640	18.6952	18.6939	18.7269	18.6733
	5	20.7606	20.1720	21.0018	20.6099	20.7340	**21.7326**
Test 05	2	12.8459	12.8459	12.8459	12.7748	12.8459	12.8459
	3	17.7525	17.7530	17.7525	17.4752	17.7482	**17.7525**
	4	19.2622	19.0092	18.9981	18.9504	18.9725	**19.2697**

Table 5 (continued)

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
	5	20.1705	20.1240	20.4218	20.0308	20.1818	**20.6769**
Test 06	2	10.0089	11.3063	10.6502	9.9340	12.6206	10.0089
	3	16.1373	16.6054	16.8058	14.2333	16.1392	**16.8165**
	4	17.2536	17.5048	16.9434	15.9115	16.4504	17.0892
	5	18.6747	19.6095	19.4061	18.5579	18.2094	**21.3203**
Test 07	2	13.9575	13.9540	13.9557	13.9528	13.9557	13.9540
	3	17.6730	17.6652	17.6681	17.6775	17.6741	17.6743
	4	18.9042	18.8782	18.8801	18.8750	18.8761	**18.9149**
	5	19.8936	19.7562	19.8538	19.7559	19.7142	**19.9672**
Test 08	2	15.3116	15.3116	15.3116	15.3116	15.3080	**15.3116**
	3	19.1258	19.1282	18.9080	19.1310	19.1266	19.1258
	4	19.5037	19.4326	19.4168	19.2724	19.3998	**19.6154**
	5	20.0871	20.0994	20.6185	19.5154	19.6849	20.6399
Test 09	2	16.0316	16.0316	16.0316	16.0320	16.0316	16.0316
	3	17.1975	17.1961	17.1968	17.1776	17.1922	17.1961
	4	19.9788	19.6102	19.8321	19.3824	19.9990	**20.1072**
	5	20.7528	20.7228	20.6128	19.8072	20.4033	**21.0635**

Table 6 SSIM segmentation effect table for each algorithm

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
Test 01	2	0.4722	0.4773	0.3640	0.2733	0.3762	0.2647
	3	0.6378	0.5795	0.6810	0.6808	0.6808	**0.6812**
	4	0.7083	0.6996	0.7233	0.7220	0.7231	**0.7234**
	5	0.7621	0.6876	0.7653	0.7466	0.7906	**0.8074**
Test 02	2	0.5707	0.5288	0.5702	0.5697	0.5703	0.5703
	3	0.6557	0.6401	0.6764	0.6750	0.6853	0.6809
	4	0.7099	0.6654	0.7106	0.6811	0.7256	0.7209
	5	0.7515	0.7067	0.7234	0.6897	0.7688	**0.7577**
Test 03	2	0.5072	0.4762	0.6196	0.6090	0.6196	**0.6196**
	3	0.6035	0.5885	0.6691	0.6716	0.6719	0.6719
	4	0.6897	0.6888	0.7991	0.7758	0.7995	**0.8004**
	5	0.7393	0.7364	0.8431	0.8403	0.8460	**0.8467**
Test 04	2	0.7082	0.5923	0.7153	0.7139	0.7156	**0.7159**
	3	0.7751	0.6285	0.7320	0.7308	0.7605	0.7315
	4	0.8303	0.7882	0.8334	0.8328	0.8331	0.8330
	5	0.8329	0.8488	0.8914	0.8839	0.8885	**0.8931**
Test 05	2	0.5071	0.4052	0.2888	0.2822	0.3494	0.2856
	3	0.6014	0.5590	0.6380	0.6189	0.6376	0.6379
	4	0.7135	0.6013	0.7110	0.7074	0.7130	**0.7269**
	5	0.7832	0.7234	0.7511	0.7523	0.7603	0.7612
Test 06	2	0.2896	0.3535	0.0784	0.0193	0.1702	0.0252
	3	0.5204	0.5207	0.5367	0.4208	0.5889	**0.5899**
	4	0.6923	0.6338	0.6187	0.5540	0.6583	0.6102
	5	0.7211	0.7039	0.7183	0.6444	0.7995	0.7347
Test 07	2	0.5331	0.5183	0.4760	0.4763	0.4757	0.4757
	3	0.6736	0.6377	0.6927	0.6916	0.6924	**0.6932**
	4	0.7475	0.6843	0.7321	0.7319	0.7317	0.7330
	5	0.7475	0.7403	0.7524	0.7538	0.7508	**0.7603**
Test 08	2	0.6653	0.6369	0.6696	0.6692	0.6696	**0.6696**
	3	0.7622	0.7348	0.7933	0.7937	0.7904	**0.7937**
	4	0.8093	0.7647	0.8037	0.7954	0.8090	**0.8095**
	5	0.8291	0.8016	0.8484	0.8399	0.8582	**0.8602**
Test 09	2	0.6598	0.6130	0.6553	0.6550	0.6554	0.6554
	3	0.7233	0.6636	0.7130	0.7122	0.7144	0.7129
	4	0.7644	0.7346	0.8050	0.8026	0.8003	**0.8051**
	5	0.7765	0.7687	0.8164	0.7954	0.8214	**0.8229**
F	index	MRFO [57]	SSA [52]	ISSA [49]	HHO-DE [7]	CPSOGSA [41]	MSMRFO
Test 01	2	0.2647	0.2647	0.2647	0.2730	0.3278	0.2647
	3	0.6812	0.6812	0.6812	0.6806	0.6812	**0.6812**
	4	0.7234	0.7232	0.7234	0.7226	0.7233	**0.7234**
	5	0.7712	0.7504	0.7760	0.7674	0.7681	**0.8074**
Test 02	2	0.5703	0.5703	0.5703	0.5702	0.5703	0.5703
	3	0.6801	0.6910	0.6791	0.6453	0.6779	0.6809
	4	0.7148	0.6992	0.6980	0.6670	0.7029	0.7209
	5	0.7479	0.7379	0.7236	0.6963	0.7103	**0.7577**
Test 03	2	0.6196	0.6196	0.6095	0.6095	0.6196	**0.6196**
	3	0.6720	0.6721	0.6720	0.6666	0.6698	0.6719
	4	0.8004	0.7961	0.8004	0.7778	0.7918	**0.8004**
	5	0.8450	0.8385	0.8431	0.8301	0.8361	**0.8467**
Test 04	2	0.7156	0.7156	0.7156	0.7150	0.7156	**0.7159**
	3	0.7316	0.7315	0.7316	0.7323	0.7325	0.7315
	4	0.8331	0.8336	0.8342	0.8361	0.8355	0.8330
	5	0.8833	0.8685	0.8898	0.8812	0.8830	**0.8931**
Test 05	2	0.2856	0.2856	0.2856	0.2792	0.2856	0.2856
	3	0.6379	0.6379	0.6379	0.6193	0.6376	0.6379
	4	0.7249	0.7126	0.7122	0.7108	0.7111	**0.7269**

Table 6 (continued)

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
	5	0.7559	0.7564	0.7663	0.7487	0.7594	0.7612
Test 06	2	0.0252	0.1325	0.0779	0.0159	0.2419	0.0252
	3	0.5335	0.5731	0.5890	0.3761	0.5342	**0.5899**
	4	0.6223	0.6298	0.5998	0.5143	0.5606	0.6102
	5	0.6883	0.7307	0.7216	0.6788	0.6735	0.7347
Test 07	2	0.4760	0.4757	0.4758	0.4755	0.4758	0.4757
	3	0.6931	0.6915	0.6920	0.6922	0.6931	**0.6932**
	4	0.7325	0.7319	0.7318	0.7323	0.7319	0.7330
	5	0.7573	0.7519	0.7560	0.7533	0.7510	**0.7603**
Test 08	2	0.6696	0.6696	0.6696	0.6696	0.6695	**0.6696**
	3	0.7927	0.7927	0.7883	0.7935	0.7930	**0.7937**
	4	0.8089	0.8045	0.8054	0.7966	0.8035	**0.8095**
	5	0.8526	0.8518	0.8570	0.8390	0.8463	**0.8602**
Test 09	2	0.6554	0.6554	0.6554	0.6553	0.6554	0.6554
	3	0.7130	0.7130	0.7130	0.7128	0.7130	0.7129
	4	0.8050	0.7910	0.7994	0.7866	0.8051	**0.8051**
	5	0.8179	0.8156	0.8151	0.7938	0.8105	**0.8229**

Table 7 FSIM segmentation effect table for each algorithm

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
Test 01	2	0.7148	0.7032	0.6850	0.6856	0.7009	0.6857
	3	0.7791	0.7506	0.7902	0.7899	0.7901	**0.7903**
	4	0.8153	0.8046	0.8301	0.8288	0.8301	**0.8303**
	5	0.8393	0.8053	0.8580	0.8470	0.8732	**0.8663**
Test 02	2	0.7406	0.7253	0.7408	0.7406	0.7409	**0.7409**
	3	0.7958	0.7848	0.8140	0.8133	0.8191	0.8170
	4	0.8281	0.8039	0.8346	0.8153	0.8462	0.8397
	5	0.8560	0.8293	0.8437	0.8226	0.8733	**0.8664**
Test 03	2	0.6801	0.6750	0.7162	0.7163	0.7162	0.7162
	3	0.7519	0.7395	0.7756	0.7765	0.7755	0.7753
	4	0.7853	0.7793	0.8419	0.8323	0.8426	**0.8434**
	5	0.8213	0.8159	0.8787	0.8772	0.8808	**0.8813**
Test 04	2	0.8454	0.7788	0.8476	0.8468	0.8477	**0.8478**
	3	0.8734	0.8077	0.8708	0.8708	0.8806	0.8703
	4	0.9010	0.8849	0.9235	0.9238	0.9234	0.9231
	5	0.9029	0.9153	0.9480	0.9454	0.9471	**0.9514**
Test 05	2	0.7780	0.7318	0.6869	0.6836	0.7150	0.6857
	3	0.8233	0.7995	0.8429	0.8374	0.8431	**0.8431**
	4	0.8741	0.8246	0.8849	0.8831	0.8860	**0.8912**
	5	0.9021	0.8759	0.9055	0.9051	0.9093	**0.9116**
Test 06	2	0.6605	0.6753	0.6092	0.5810	0.6359	0.5973
	3	0.7240	0.7222	0.7418	0.7026	0.7565	**0.7565**
	4	0.7950	0.7701	0.7676	0.7429	0.7991	0.7690
	5	0.8137	0.8028	0.8230	0.7807	0.8672	0.8155
Test 07	2	0.8144	0.8028	0.8043	0.8044	0.8042	0.8042
	3	0.8740	0.8559	0.8854	0.8851	0.8852	**0.8855**
	4	0.8978	0.8759	0.9098	0.9098	0.9095	**0.9098**
	5	0.9011	0.8954	0.9209	0.9216	0.9201	**0.9236**
Test 08	2	0.7601	0.7354	0.7656	0.7653	0.7656	**0.7656**
	3	0.8241	0.7962	0.8455	0.8451	0.8450	**0.8458**
	4	0.8444	0.8238	0.8540	0.8455	0.8589	**0.8594**
	5	0.8645	0.8436	0.8910	0.8797	0.8925	**0.8924**
Test 09	2	0.7502	0.7240	0.7528	0.7526	0.7528	**0.7528**
	3	0.7836	0.7612	0.8090	0.8086	0.8084	**0.8090**
	4	0.8170	0.8066	0.8435	0.8416	0.8428	**0.8457**
	5	0.8320	0.8197	0.8599	0.8468	0.8658	0.8665
F	index	MRFO [57]	SSA [52]	ISSA [49]	HHO-DE [7]	CPSOGSA [41]	MSMRFO
Test 01	2	0.6857	0.6857	0.6857	0.6855	0.6852	0.6857
	3	0.7903	0.7903	0.7903	0.7899	0.7903	**0.7903**
	4	0.8303	0.8301	0.8303	0.8294	0.8303	**0.8303**
	5	0.8615	0.8493	0.8646	0.8593	0.8599	**0.8663**
Test 02	2	0.7409	0.7409	0.7409	0.7409	0.7409	**0.7409**
	3	0.8166	0.8217	0.8153	0.7934	0.8142	0.8170
	4	0.8379	0.8282	0.8271	0.8063	0.8307	0.8397
	5	0.8606	0.8524	0.8434	0.8244	0.8352	**0.8664**
Test 03	2	0.7162	0.7162	0.7163	0.7163	0.7162	0.7162
	3	0.7754	0.7755	0.7754	0.7708	0.7759	0.7753
	4	0.8434	0.8410	0.8433	0.8321	0.8388	**0.8434**
	5	0.8802	0.8750	0.8788	0.8716	0.8733	**0.8813**
Test 04	2	0.8477	0.8477	0.8477	0.8474	0.8477	**0.8478**
	3	0.8703	0.8703	0.8704	0.8714	0.8710	0.8703
	4	0.9232	0.9232	0.9233	0.9236	0.9235	0.9231
	5	0.9456	0.9402	0.9469	0.9431	0.9447	**0.9514**
Test 05	2	0.6857	0.6857	0.6857	0.6822	0.6857	0.6857
	3	0.8431	0.8431	0.8431	0.8381	0.8430	**0.8431**
	4	0.8917	0.8856	0.8854	0.8843	0.8849	**0.8912**

Table 7 (continued)

F	index	GWO [35]	ABC [30]	BSA [32]	WOA [34]	PSO [31]	MSMRFO
	5	0.9083	0.9074	0.9113	0.9038	0.9089	**0.9116**
Test 06	2	0.5973	0.6231	0.6075	0.5741	0.6533	0.5973
	3	0.7403	0.7518	0.7557	0.6915	0.7417	**0.7565**
	4	0.7784	0.7763	0.7624	0.7277	0.7487	0.7690
	5	0.8136	0.8290	0.8230	0.7950	0.8014	0.8155
Test 07	2	0.8043	0.8042	0.8042	0.8042	0.8042	0.8042
	3	0.8855	0.8850	0.8852	0.8852	0.8855	**0.8855**
	4	0.9096	0.9097	0.9096	0.9099	0.9097	0.9098
	5	0.9226	0.9204	0.9221	0.9214	0.9202	**0.9236**
Test 08	2	0.7656	0.7656	0.7656	0.7656	0.7655	**0.7656**
	3	0.8458	0.8458	0.8441	0.8454	0.8457	**0.8458**
	4	0.8581	0.8545	0.8545	0.8478	0.8529	**0.8594**
	5	0.8926	0.8916	0.8922	0.8824	0.8878	**0.8924**
Test 09	2	0.7528	0.7528	0.7528	0.7528	0.7528	**0.7528**
	3	0.8090	0.8090	0.8090	0.8088	0.8090	**0.8090**
	4	0.8436	0.8397	0.8421	0.8362	0.8435	**0.8457**
	5	0.8613	0.8608	0.8577	0.8429	0.8552	0.8665

Table 8 Ranking results of MSMRFO with other algorithms

F	index	GWO	ABC	BSA	WOA	PSO	MSMRFO
Test 01	2	2.8000	3.0111	3.7500	4.0333	3.3222	4.0833
	3	3.2444	5.0778	3.2333	3.0611	3.3611	**3.0222**
	4	3.6667	4.5222	2.9000	4.0278	3.3722	**2.5111**
	5	3.7667	5.1222	3.1056	4.1889	2.4500	**2.3667**
Test 02	2	3.2667	4.7444	3.2111	3.4889	3.1444	**3.1444**
	3	3.7667	4.4778	3.3278	3.9056	2.4778	**3.0444**
	4	3.3444	4.7444	3.1667	4.6889	2.4056	2.6500
	5	2.8444	4.2444	4.0889	5.2778	2.5222	**2.0222**
Test 03	2	4.7111	5.1333	2.7444	2.9222	2.7444	**2.7444**
	3	4.4111	4.8333	3.0333	2.8500	2.9333	2.9389
	4	5.2667	5.2222	2.4833	3.9556	2.1389	**1.9333**
	5	5.3667	5.5444	2.8111	3.4333	1.9722	**1.8722**
Test 04	2	2.7444	5.3667	3.1889	3.7000	3.0333	2.9667
	3	2.1556	4.8444	3.5611	4.0389	2.8333	3.5667
	4	3.9333	4.7000	3.0500	3.2500	3.0667	**3.0000**
	5	4.8889	4.7000	2.7778	2.8278	2.9056	2.9000
Test 05	2	1.8667	2.8333	4.1056	4.4722	3.5333	4.1889
	3	3.8333	4.5222	2.7389	4.1778	2.9222	2.8056
	4	3.0556	4.9333	3.4500	4.1111	3.4444	**2.0056**
	5	3.0000	4.2556	3.7778	4.0667	3.0000	**2.9000**
Test 06	2	2.6889	2.5000	3.9278	4.6556	3.9722	3.2556
	3	3.5889	3.7111	3.0111	4.1944	3.2500	3.2444
	4	2.6333	3.1333	3.8667	4.8222	2.5556	3.9889
	5	3.6111	3.6778	3.4667	4.7333	1.7667	3.7444
Test 07	2	2.4000	3.3000	3.7722	3.6611	3.9333	3.9333
	3	5.2000	4.7111	2.7500	3.0556	2.7222	**2.5611**
	4	3.5000	5.0222	3.2056	3.2389	3.2833	**2.7500**
	5	4.4000	4.7444	3.1778	3.0722	3.6167	**1.9889**
Test 08	2	3.3444	4.6222	3.1556	3.5667	3.1556	**3.1556**
	3	4.6444	5.1000	2.6611	2.8111	2.9222	2.8611
	4	3.0667	4.6556	3.2722	4.6111	2.5389	2.8556
	5	3.6556	4.7333	3.3667	4.2111	2.3222	**2.7111**
Test 09	2	3.8667	5.3556	2.8500	3.3611	2.7833	**2.7833**
	3	3.3222	4.9000	3.1833	3.5778	3.0056	**3.0111**
	4	4.9222	5.5889	2.4333	3.1611	2.4944	**2.4000**
	5	4.4778	5.0222	2.6611	4.2778	2.4167	**2.1444**

Table 9 Ranking results of MSMRFO and algorithms in recent years

F	index	MRFO	SSA	ISSA	HHO-DE	CPSOGSA	MSMRFO
Test 01	2	1.0889	3.9611	3.9611	4.3389	3.6889	3.9611
	3	3.4389	3.4389	3.4389	3.8889	3.4389	**3.3556**
	4	2.8889	3.6722	3.0444	4.2222	3.1500	4.0222
	5	2.8889	4.4778	3.2222	3.9889	3.8111	**2.6111**
Test 02	2	3.4944	3.4944	3.4944	3.5278	3.4944	**3.4944**
	3	3.2889	2.2222	3.2778	4.7389	3.9000	3.5722
	4	2.8833	3.3278	3.9278	5.0500	3.3500	**2.4611**
	5	2.5000	3.1889	3.8833	4.9444	4.5389	**1.9444**
Test 03	2	3.4889	3.4889	3.5222	3.5222	3.4889	**3.4889**
	3	3.2667	3.1778	4.2556	3.7111	3.2556	3.3333
	4	3.0500	2.9500	3.0833	5.7333	3.3333	**2.8500**
	5	2.8111	3.5444	3.2111	4.5667	4.6389	**2.2278**
Test 04	2	3.4222	3.4556	3.4556	3.8556	3.4556	**3.3556**
	3	3.5556	3.5611	3.5167	3.4444	3.4056	3.5167
	4	3.5611	3.7389	3.5833	3.2556	3.1222	3.7389
	5	3.3111	4.3222	3.0667	3.5222	3.3889	3.3889
Test 05	2	3.3667	3.3667	3.3667	4.1667	3.3667	**3.3667**
	3	3.3389	2.3889	3.3389	5.3333	3.2611	3.3389
	4	1.8056	4.1944	4.2722	4.5056	4.4611	**1.7611**
	5	3.1944	4.5778	2.7611	4.1444	3.3556	**2.9667**
Test 06	2	3.3667	3.1833	3.4333	5.1833	2.4667	3.3667
	3	3.6389	3.2167	3.5222	4.9000	2.2111	3.5111
	4	2.8333	2.8389	3.3944	4.4333	4.0333	3.4667
	5	3.7056	2.7278	2.9333	4.3778	3.7944	3.4611
Test 07	2	3.3500	3.5500	3.4500	3.6500	3.4500	3.5500
	3	3.2500	3.8333	3.8333	3.7333	3.1889	**3.1611**
	4	3.2333	3.7000	3.6833	3.5389	3.8000	**3.0444**
	5	2.6444	4.4000	3.2444	3.9111	4.6444	**2.1556**
Test 08	2	3.4833	3.4833	3.4833	3.4833	3.5833	**3.4833**
	3	3.5444	3.6389	3.6611	3.1500	3.4722	3.5333
	4	2.3167	3.5111	3.5222	4.9611	4.0056	2.6833
	5	2.9778	3.1833	2.9778	4.9889	3.9722	**2.9000**
Test 09	2	3.4889	3.4889	3.4889	3.5556	3.4889	**3.4889**
	3	3.2444	3.3056	3.2778	4.3056	3.5056	3.3611
	4	3.1333	3.7333	3.8000	4.0222	3.1000	3.2111
	5	2.9667	2.9556	3.3000	5.3833	3.6889	**2.7056**

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos.62002046,62006106),The Project Supported by Zhejiang Provincial Natural Science Foundation of China(No.LQ21F020005), Basic public welfare research program of Zhejiang Province(No.LGG18E050011).

Data availability The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

References

1. Abd El Aziz M, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256. https://doi.org/10.1016/j.eswa.2017.04.023
2. Abd Elaziz M, Yousri D, Al-qaness MA, AbdelAty AM, Radwan AG, Ewees AA (2021) A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation. Eng Appl Artif Intell 98:104105. https://doi.org/10.1016/j.engappai.2020.104105
3. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13(6):3066–3091. https://doi.org/10.5555/2467341.2467507
4. Akdag O, Yeroglu C (2021) Optimal directional overcurrent relay coordination using MRFO algorithm: A case study of adaptive protection of the distribution network of the Hatay province of Turkey. Electr Power Syst Res 192:106998. https://doi.org/10.1016/j.epsr.2020.106998
5. Alturki FA, Farh MH, Al-Shamma'a AA, AlSharabi K (2020) Techno-economic optimization of small-scale hybrid energy systems using manta ray foraging optimizer. Electronics 9(12):2045. https://doi.org/10.3390/electronics9122045
6. Aly M, Rezk H (2021) A MPPT based on optimized FLC using manta ray foraging optimization algorithm for thermo-electric generation systems. Int J Energy Res 45(9):13897–13910. https://doi.org/10.1002/er.6728
7. Bao X, Jia H, Lang C (2019) A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation. IEEE Access 7:76529–76546. https://doi.org/10.1109/ACCESS.2019.2921545
8. Ben UC, Akpan AE, Mbonu CC, Ebong ED (2021) Novel methodology for interpretation of magnetic anomalies due to two-dimensional dipping dikes using the Manta ray foraging optimization. J Appl Geophys 192:104405. https://doi.org/10.1016/j.jappgeo.2021.104405
9. Das S, Suganthan PN (2010) Differential evolution: A survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31. https://doi.org/10.1109/TEVC.2010.2059031
10. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30. https://doi.org/10.5555/1248547.1248548
11. Ekinci S, Izci D, Hekimoğlu B (2021) Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm. Arab J Sci Eng 46(2):1395–1409. https://doi.org/10.1007/s13369-020-05050-z
12. Elmaadawy K, Abd Elaziz M, Elsheikh AH, Moawad A, Liu B, Lu S (2021) Utilization of random vector functional link integrated with manta ray foraging optimization for effluent prediction of wastewater treatment plant. J Environ Manag 298:113520. https://doi.org/10.1016/j.jenvman.2021.113520
13. Erdmann H, Wachs-Lopes G, Gallao C, Ribeiro MP, Rodrigues PS (2015) A study of a firefly meta-heuristics for multithreshold image segmentation. In Developments in medical image processing and computational vision. Springer, Cham. pp. 279–295 https://doi.org/10.1007/978-3-319-13407-9_17
14. Fathy A, Rezk H, Yousri D (2020) A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm[J]. Sol Energy 207:305–316. https://doi.org/10.1016/j.solener.2020.06.108
15. Fayad H, Hatt M, Visvikis D (2015) PET functional volume delineation using an Ant colony segmentation approach. (2015):1745–1745. https://jnm.snmjournals.org/content/56/supplement_3/1745.short
16. Fogel DB, Atmar JW (1990) Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems. Biol Cybern 63(2):111–114. https://doi.org/10.1007/BF00203032

17. Ghosh KK, Guha R, Bera SK, Kumar N, Sarkar R (2021) S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem. Neural Comput & Applic 33(17): 11027–11041. https://doi.org/10.1007/s00521-020-05560-9
18. Haralick RM, Shapiro LG (1985) Image segmentation techniques. Comput Vis Graph Image Process 29(1): 100–132. https://doi.org/10.1016/S0734-189X(85)90153-7
19. Hassan MH, Houssein EH, Mahdy MA, Kamel S (2021) An improved manta ray foraging optimizer for cost-effective emission dispatch problems. Eng Appl Artif Intell 100:104155. https://doi.org/10.1016/j.engappai.2021.104155
20. Hemeida MG, Alkhalaf S, Mohamed AAA, Ibrahim AA, Senjyu T (2020) Distributed generators optimization based on multi-objective functions using manta rays foraging optimization algorithm (MRFO). Energies 13(15):3847. https://doi.org/10.3390/en13153847
21. Hemeida MG, Ibrahim AA, Mohamed AAA, Alkhalaf S, El-Dine AMB (2021) Optimal allocation of distributed generators DG based Manta ray foraging optimization algorithm (MRFO). Ain Shams Eng J 12(1):609–619. https://doi.org/10.1016/j.asej.2020.07.009
22. Higashi N, Iba H (2003, April) Particle swarm optimization with Gaussian mutation. In: Proceedings of the 2003 IEEE swarm intelligence symposium. SIS'03 (cat. No. 03EX706). IEEE. pp. 72-79 https://doi.org/10.1109/SIS.2003.1202250
23. Houssein EH, Zaki GN, Diab AAZ, Younis EM (2021) An efficient Manta ray foraging optimization algorithm for parameter extraction of three-diode photovoltaic model. Comput Electr Eng 94:107304. https://doi.org/10.1016/j.compeleceng.2021.107304
24. Houssein EH, Ibrahim IE, Neggaz N, Hassaballah M, Wazery YM (2021) An efficient ECG arrhythmia classification method based on Manta ray foraging optimization. Expert Syst Appl 181:115131. https://doi.org/10.1016/j.eswa.2021.115131
25. Houssein EH, Emam MM, Ali AA (2021) Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images. Neural Comput & Applic 33(24):16899–16919. https://doi.org/10.1007/s00521-021-06273-3
26. Islam MJ, Luo P, Sattar J (2020) Simultaneous enhancement and super-resolution of underwater imagery for improved visual perception. arXiv preprint arXiv:2002.01155. https://doi.org/10.48550/arXiv.2002.01155
27. Jena B, Naik MK, Panda R, Abraham A (2021) Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta ray foraging optimization. Eng Appl Artif Intell 103:104293. https://doi.org/10.1016/j.engappai.2021.104293
28. Jia H, Lang C, Oliva D, Song W, Peng X (2019) Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation. Remote Sens 11(12):1421. https://doi.org/10.3390/rs11121421
29. Jin H, Li Y, Xing B, Wang L (2016) A geometric image segmentation method based on a bi-convex, fuzzy, variational principle with teaching-learning optimization. J Intell Fuzzy Syst 31(6):3075–3081. https://doi.org/10.3233/JIFS-169193
30. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697. https://doi.org/10.1016/j.asoc.2007.05.007
31. Kennedy J, Eberhart RC (1997, October) A discrete binary version of the particle swarm algorithm. In: 1997 IEEE international conference on systems, man, and cybernetics. Computational cybernetics and simulation. IEEE. (Vol. 5, pp. 4104-4108) https://doi.org/10.1109/ICSMC.1997.637339
32. Meng XB, Gao XZ, Lu L, Liu Y, Zhang H (2016) A new bio-inspired optimisation algorithm: bird swarm algorithm. J Exp Theor Artif Intell 28(4):673–687. https://doi.org/10.1080/0952813X.2015.1042530
33. Micev M, Ćalasan M, Ali ZM, Hasanien HM, Aleem SHA (2021) Optimal design of automatic voltage regulation controller using hybrid simulated annealing–Manta ray foraging optimization algorithm. Ain Shams Eng J 12(1):641–657. https://doi.org/10.1016/j.asej.2020.07.010
34. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
35. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
36. Pare S, Kumar A, Bajaj V, Singh GK (2016) A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve. Appl Soft Comput 47:76–102. https://doi.org/10.1016/j.asoc.2016.05.040
37. Peng H, Zhu W, Deng C, Wu Z (2021) Enhancing firefly algorithm with courtship learning. Inf Sci 543:18–42. https://doi.org/10.1016/j.ins.2020.05.111
38. Peng H, Zeng Z, Deng C, Wu Z (2021) Multi-strategy serial cuckoo search algorithm for global optimization. Knowl-Based Syst 214:106729. https://doi.org/10.1016/j.knosys.2020.106729

39. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315. https://doi.org/10.1016/j.cad.2010.12.015
40. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
41. Rather SA, Bala PS (2021) Constriction coefficient based particle swarm optimization and gravitational search algorithm for multilevel image thresholding. Expert Syst 38(7):e12717. https://doi.org/10.1111/exsy.12717
42. Saleh S, Kalyankar NV, Khamitkar SD (2010) Image segmentation by using threshold techniques. J Comput 2:2151–9617. https://arxiv.53yu.com/abs/1005.4020
43. Salgotra R, Singh U (2019) The naked mole-rat algorithm. Neural Comput & Applic 31(12):8837–8857. https://doi.org/10.1007/s00521-019-04464-7
44. Sayed GI, Solyman M, Hassanien AE (2019) A novel chaotic optimal foraging algorithm for unconstrained and constrained problems and its application in white blood cell segmentation. Neural Comput & Applic 31(11):7633–7664. https://doi.org/10.1007/s00521-018-3597-8
45. Shaheen AM, Ginidi AR, El-Sehiemy RA, Ghoneim SS (2020) Economic power and heat dispatch in cogeneration energy systems using manta ray foraging optimizer. IEEE Access 8:208281–208295. https://doi.org/10.1109/ACCESS.2020.3038740
46. Upadhyay P, Chhabra JK (2020) Kapur's entropy based optimal multilevel image segmentation using crow search algorithm. Appl Soft Comput 97:105522. https://doi.org/10.1016/j.asoc.2019.105522
47. Wang S, Jia H, Peng X (2020) Modified salp swarm algorithm based multilevel thresholding for color image segmentation. Math Biosci Eng 17(1):700–724. https://doi.org/10.3934/mbe.2020036
48. Whitley D (1994) Genetic algorithm tutorial. Stat Comput 4(2):65–85. https://doi.org/10.1007/BF00175354
49. Xin Lv, Xiaodong M, Jun Z (2021) Multi-threshold image segmentation based on improved sparrow search algorithm. Syst Eng Electron Technol 43(2):10. https://doi.org/10.12305/j.issn.1001-506X.2021.02.05
50. Xin LV, Xiaodong M, Jun Z, Zhen W (2021) Chaotic sparrow search optimization algorithm. J Beijing Univ Aeronaut Astronaut 47(8):1712–1720. https://doi.org/10.13700/j.bh.1001-5965.2020.0298
51. Xu H, Song H, Xu C, Wu X, Yousefi N (2020) Exergy analysis and optimization of a HT-PEMFC using developed Manta ray foraging optimization algorithm. Int J Hydrog Energy 45(55):30932–30941. https://doi.org/10.1016/j.ijhydene.2020.08.053
52. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8(1):22–34. https://doi.org/10.1080/21642583.2019.1708830
53. Yang XS, Deb S (2009, December) Cuckoo search via Lévy flights. In: 2009 world congress on nature & biologically inspired computing (NaBIC). IEEE. pp. 210-214 https://doi.org/10.1109/NABIC.2009.5393690
54. Yang Z, Wu A (2020) A non-revisiting quantum-behaved particle swarm optimization based multilevel thresholding for image segmentation. Neural Comput & Applic 32(16):12011–12031. https://doi.org/10.1007/s00521-019-04210-z
55. Yu Y, Gao S, Wang Y, Cheng J, Todo Y (2018) ASBSO: an improved brain storm optimization with flexible search length and memory-based selection. IEEE Access 6:36977–36994. https://doi.org/10.1109/ACCESS.2018.2852640
56. Zhao D, Liu L, Yu F, Heidari AA, Chen H (2020) Ant colony optimization with horizontal and vertical crossover search: fundamental visions for multi-threshold image segmentation. Expert Syst Appl, http://aliasgharheidari.com:114122. https://doi.org/10.1016/j.eswa.2020.114122
57. Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300. https://doi.org/10.1016/j.engappai.2019.103300
58. Zhou Y, Yang X, Ling Y, Zhang J (2018) Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation. Multimed Tools Appl 77(18):23699–23727. https://doi.org/10.1007/s11042-018-5637-x