IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 9, 2020, accepted September 17, 2020, date of publication September 22, 2020, date of current version October 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025833

# A Scheme of Color Image Multithreshold Segmentation Based on Improved Moth-Flame Algorithm

**TRONG-THE NGUYEN[1,2], HONG-JIANG WANG[1], THI-KIEN DAO[1], JENG-SHYANG PAN[1,5], TRUONG-GIANG NGO[3], AND JIE YU[4]**

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China
[2]Department of Information Technology, Haiphong University of Management and Technology, Haiphong 180000, Vietnam
[3]Faculty of Computer Science and Engineering, Thuyloi University, Hanoi 116705, Vietnam
[4]College of Mechanical and Automotive Engineering, Fujian University of Technology, Fuzhou 350118, China
[5]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

Corresponding author: Thi-Kien Dao (jvnkien@gmail.com)

This work is supported in part by the National Natural Science Foundation of China (61872085), in part by the Natural Science Foundation of Fujian Province (2018J01638), and in part by the Fujian Provincial Department of Science and Technology (2018Y3001).

**ABSTRACT** A recently developed swarm intelligence algorithm by studying the natural moth's biological behavior is called Moth-Flame Optimization (MFO). The advantages of MFO conclude a simple structure and a robust selection capability. Still, it is easy to be trapped falling into optimal local, and slow search converges. This study suggests a new process improving MFO by hybridizing Lévy flight and logarithmic functions for its formula of flame updating to enhance the optimization performance of the algorithm. In the experimental section, a set of benchmark functions of CEC2013 and the multi threshold image segmentation are used to evaluate the proposed method performance. Compared results of the proposed methods with the different algorithms in the same condition scenarios show that the suggested approach provides better results than the various algorithms in the competitions.

**INDEX TERMS** Moth-flame algorithm, color image segmentation, multi threshold segmentation, minimum cross-entropy.

## I. INTRODUCTION

The synergy of cooperation and competition exists widely in the natural world that are essential factors for the survival of populations [1], e.g., the colony of ants [2], colonies bees [3], flocks of birds [4], even bees related to pollen flowers [5], etc [6]. The individuals are smarter when working in swarms or teams [7], the synergy much higher than the parts that, based on the cooperation, provides the inspiration for intelligent computation [8], [9]. The swarm intelligence (SI) algorithm has implemented synergy by bionic approaches, such as the Genetic Algorithm (GA) [10], Ants and Bees colonies (ACO) [2] and (ABC) [3], Particle Swarm Optimization (PSO) [11], Grey Wolf Optimization (GWO) [12], Cuckoo Search (CS) [13], Differential Evolution (DE) [14], and Parallel particle swarm optimization (PPSO) [15].

Moth-Flame Optimization (MFO) [16] is a new swarm intelligence bionic algorithm that taken the inspiration from natural moth behavior. Due to its excellent performance, the algorithm has been widely used in engineering fields [17], e.g., a confined aquifer parameter inversion, Muskingum model parameter optimization [18], network flow prediction [19], and power system optimal power flow calculation [20].

Moreover, image segmentation is a key step in image analyzing and processing that transforms the original image into a more abstract and compact form, which makes it possible for higher-level image analysis [21]. The expansion of the application field of imaging equipment, the application field of image segmentation, is also expanding, such as in the field of medicine, intelligent transportation, video monitoring, industrial production, and so on [22]. The quality of segmentation directly affects the accuracy of feature extraction, measurement, image recognition, and understanding in image analysis [23].

Among image segmentation algorithms, the threshold segmentation algorithm is widely used because of its simple calculation, high efficiency, and fast speed [24]. The traditional threshold segmentation method is beneficial for the single threshold segmentation [25]. However, for the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

multi threshold segmentation, the traditional multi threshold segmentation algorithm has suffered from increasing complexity computation segmentation time rapidly with the large threshold number [26]. Because it uses the exhaustive search for the best threshold, with the increasing number of thresholds, the amount of calculation will increase rapidly, the operation time becomes more prolonged, and the operation speed becomes slower [27]. The process of searching for the optimal threshold for a given image can be regarded as a constrained optimization problem. The optimal threshold can be obtained by optimizing the objective function. The minimum cross-entropy threshold segmentation is to select the optimal threshold combination based on the minimum cross-entropy [28].

However, with the increasing number of thresholds, the computational complexity also increases exponentially, which directly affects the efficiency of image segmentation [29]. Therefore, a multi-level threshold problem is the extension of the optimum threshold for image segmentation by maximizing the interclass variance that becomes very time-consuming because a large number of iterations required to calculate each class mean and cumulative probability. The SI algorithm is one of the excellent ways to deal with the complicated threshold selection problem by learning from the bionic algorithm to improve the optimization efficiency [30], [31]. A recently developed SI algorithm, MFO has the advantages of a simple structure and a robust selection capability [16], [17]. Still, it is easy to be trapped falling into optimal local, and slow search converges [32]. For these reasons of weaknesses points when dealing with complicated problems, MFO has been further notice paid attention to changing equations and parameters for improving the performance functionality, e.g., the enhanced MFO with mutation strategy (EMFO) [33], chaotic mutative MFO (CMFO) [34]. Especially, MFO had been combined with the Lévy flight for function optimization and engineering design problems [35]. MFO has been applied to the practical issues, e.g., Intelligent identification of facial expressions through the MFO [36]; An unsmooth economic situation emissions dispatch issues by enhanced MFO [37]; and Parameter identification of single-phase inverter with improved MFO [38]. However, as the No-Free-Lunch (NFL) theorem [39] for optimization said, there is a crucial question as to whether there is an optimization algorithm for solving all problems of optimization. It is aiming at these disadvantages of the MFO algorithm, this article considered to improve MFO by hybridizing Lévy flight and logarithmic functions for its formula of flame updating to enhance the optimization performance of the algorithm. The minimum cross-entropy is taken as the optimization objective function, and the improved MFO algorithm is applied to multithreshold color image segmentation. As our knowledge, besides combining MFO with the Lévy flight as the mentioned previous works, we also adapt the inertia weight and logarithmic functions while processing optimization to enhance the converge of the algorithm.

The contributions behind the proposed scheme are highlighted as follows.

- Enhancing MFO by hybridizing Lévy flight and logarithmic functions for its formula of updating flame based on a mutation probability.
- Experimenting evaluating the proposed method performance through testing the selected benchmark functions.
- Applying the new proposed algorithm to solve to the complex of multi threshold color image segmentation.

The rest of the paper is arranged as follows: Section 2 reviews the conical MFO and the statement of the multi threshold color image segmentation problem as the related work. Section 3, which proposes an improvement version of MFO. Section 4 describes the improved MFO for the multi threshold color image segmentation issue. The ending is discussed as the conclusion in Section 5.

## II. RELATED WORD

In this section, a new swarm intelligence bionic algorithm called the moth-flame optimization (MFO) that inspiration taken from the moth's flight mode is reviewed in subsection A. And subsection B would report the statement of the multi-threshold color image segmentation problem. The subsections are presented in detail as follows.

### A. MOTH-FLAME OPTIMIZATION ALGORITHM

MFO is considered as a novel swarm intelligence optimization algorithm with the inspiration from the moth's flight

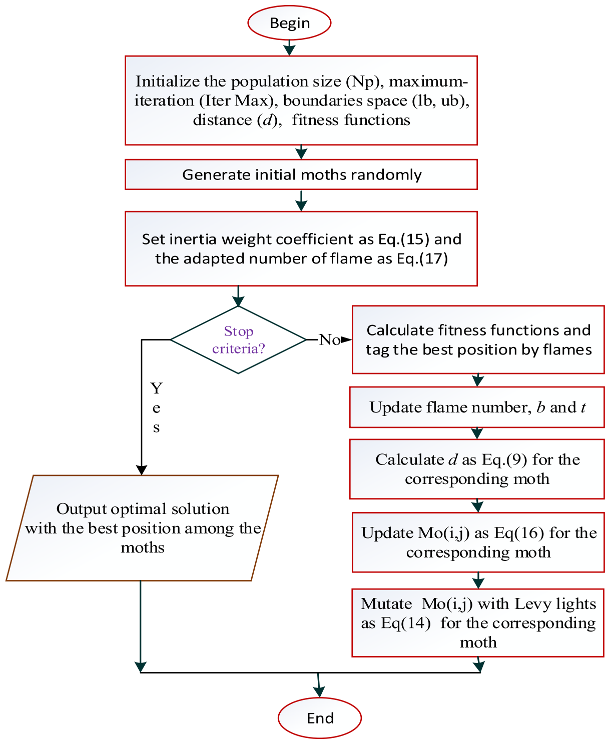


**FIGURE 1. Flowchart of the proposed IMFO.**

**TABLE 1.** The initial rang, dimension, and description of selected test benchmark functions.

| Series | Description of the problems | Dimension | Range | $f_{min}$ |
|---|---|---|---|---|
| **A** | **Unimodal benchmark functions** | | | |
| 1 | $f_1(x) = \sum_{i=1}^{n} x_i^2$ | 30 | [ -100, 100] | 0 |
| 2 | $f_2(x) = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} x_j\right)^2$ | 30 | [ -100, 100] | 0 |
| 3 | $f_3(x) = \max_i\{\lvert x_i\rvert, 1 \leq i \leq n\}$ | 30 | [ -100, 100] | 0 |
| 4 | $f_4(x) = \sum_{i=1}^{n-1}[100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30 | [ -100, 100] | 0 |
| **B** | **Multimodal benchmark functions** | | | |
| 5 | $f_5(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$ | 30 | [ -500, 500] | 0 |
| 6 | $f_6(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$ | 30 | [-5.12, 5.12] | 0 |
| 7 | $f_7(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})$ <br> - $\exp(\frac{1}{n}\cos(2\pi x_i)) + 20 + e))$ | 30 | [-32,32] | 0 |
| 8 | $f_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | 30 | [-600, 600] | 0 |
| 9 | $f_9(x) = \frac{\pi}{n}(10\sin(\pi y_1) + \sum_{i=1}^{n}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))$ <br> $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ <br> $y_i = 1 + \frac{x_i+1}{4}$ <br> $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ | 30 | [-50, 50] | 0 |
| 10 | $f_{10}(x) = 0.1(\sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2(1 + \sin^2(3\pi x_i + 1))$ <br> $+(x_n - 1)^2(1 + \sin^2(2\pi x_n))) + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | 30 | [-50, 50] | 0 |
| **C** | **Fixed-dimension multimodal benchmark functions** | | | |
| 11 | $f_{11}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_1 - a_{ij})^6})^{-1}$ | 2 | [-65,65] | 1 |
| 12 | $f_{12}(x) = \sum_{i=1}^{11}(a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4})$ | 4 | [-5,5] | 0.0003 |
| 13 | $f_{13}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ | 2 | [-5,5] | -1.0316 |
| 14 | $f_{14}(x) = (x^2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2$ <br> $+10(1 - \frac{1}{8\pi})\cos x_1 + 10$ | 2 | [-5,5] | 0.3980 |
| 15 | $f_{15}(x) = (1 + (x_1 + x_2 + 1)^2$ <br> $\times(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)$ <br> $\times(30 + (2x_1 - 3x_2)^2$ <br> $\times(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$ | 2 | [-2,2] | 3 |
| 16 | $f_{16}(x) = -\sum_{i=1}^{4} c_1 \times \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$ | 3 | [1,3] | -3.860 |
| 17 | $f_{17}(x) = -\sum_{i=1}^{4} c_1 \times \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$ | 6 | [0,1] | -3.320 |
| 18 | $f_{18}(x) = -\sum_{i=1}^{5}((X - a_i)(X - a_i)^T + c_i)^{-1}$ | 4 | [0,10] | -10.15320 |
| 19 | $f_{19}(x) = -\sum_{i=1}^{7}((X - a_i)(X - a_i)^T + c_i)^{-1}$ | 4 | [0,10] | -10.4028 |
| 20 | $f_{20}(x) = -\sum_{i=1}^{10}((X - a_i)(X - a_i)^T + c_i)^{-1}$ | 4 | [0,10] | -10.53630 |
| **D** | **Composite benchmark functions** | | | |
| 21 | $f_{21}(x) =$ <br> $f_1, \ldots, f_{10}$=Sphere function <br> $[\delta_1, \ldots, \delta_{10}] = [1, \ldots, 1]$ <br> $[\lambda_1, \ldots, \lambda_{10}] = [\frac{5}{100}, \ldots, \frac{5}{100}]$ | 10 | [-5,5] | 0 |
| 22 | $f_{22}(x) =$ <br> $f_1, \ldots, f_{10}$=Griewank function <br> $[\delta_1, \ldots, \delta_{10}] = [1, \ldots, 1]$ <br> $[\lambda_1, \ldots, \lambda_{10}] = [\frac{5}{100}, \ldots, \frac{5}{100}]$ | 10 | [-5,5] | 0 |
| 23 | $f_{23}(x) =$ <br> $f_1, \ldots, f_{10}$=Rastrigin function <br> $[\delta_1, \ldots, \delta_{10}] = [1, \ldots, 1]$ <br> $[\lambda_1, \ldots, \lambda_{10}] = [1, \ldots, 1]$ | 10 | [-5,5] | 0 |

**TABLE 2. Test results of the proposed IMFO, MFO and GWO algorithms for selected 23 functions.**

| Test Func. | MFO | | | GWO | | | IMFO | | |
|---|---|---|---|---|---|---|---|---|---|
| | Best | Mean | Std: | Best | Mean | Std: | Best | Mean | Std: |
| 1 | 2.83E-10 | 6.87E-10 | 7.01E-08 | **2.38E-12** | 8.92E-10 | 7.04E-08 | 2.79E-12 | **3.36E-11** | 1.40E-07 |
| 2 | 5.82E-03 | 1.45E+00 | 1.32E+00 | 8.42E-03 | 2.53E+00 | 3.00E+00 | 1.30E-04 | 1.57E-01 | 5.01E-04 |
| 3 | 4.07E-04 | 1.51E-01 | 2.09E-01 | 3.19E-04 | 1.45E-01 | 1.67E-01 | **1.13E-04** | 1.35E-01 | 9.10E-02 |
| 4 | 2.60E-14 | 3.10E-09 | 2.62E-04 | 9.01E-15 | 2.58E-02 | 1.87E-02 | **5.15E-16** | 6.20E-09 | 5.24E-04 |
| 5 | 4.48E-02 | 2.04E+01 | 6.66E-02 | 3.64E-02 | 2.04E+01 | 6.53E-02 | 1.66E-03 | 1.99E+01 | 8.74E-02 |
| 6 | 8.41E-02 | 2.13E+01 | 9.88E-01 | 6.01E-02 | 1.89E+01 | 2.07E+00 | 2.07E-04 | 2.49E+00 | **9.66E-01** |
| 7 | 5.68E-04 | 2.35E-01 | 1.24E-01 | 1.99E-01 | 5.86E+01 | 2.77E+01 | **1.69E-05** | 2.04E-01 | 1.52E-01 |
| 8 | 2.70E-02 | 9.32E+00 | 4.35E+00 | 4.60E-01 | 1.38E+02 | 2.97E+01 | **4.85E-04** | 5.84E+00 | 3.68E+00 |
| 9 | 3.39E-01 | 8.55E+01 | 4.53E+00 | 4.56E-01 | 1.39E+02 | 3.99E+01 | **8.36E-04** | 1.01E+01 | 8.16E-02 |
| 10 | 3.35E-01 | 8.59E+01 | 7.43E+00 | 5.71E-01 | 1.74E+02 | 3.49E+01 | 1.07E-03 | 1.28E+01 | 6.12E+00 |
| 11 | 1.62E+01 | 4.39E+03 | 3.33E+02 | 7.09E+00 | 2.59E+03 | 5.30E+02 | 8.69E-02 | 1.05E+03 | 3.49E+02 |
| 12 | 5.26E-02 | 4.55E-01 | 2.71E-01 | 5.17E-02 | 2.77E-01 | 1.48E-01 | 1.02E-01 | 1.23E-01 | -1.86E-01 |
| 13 | 4.87E-05 | 5.64E-01 | 1.14E-01 | 3.87E-03 | 1.69E+00 | 3.23E-01 | **9.37E-05** | 1.13E+00 | **2.26E-01** |
| 14 | 3.28E-01 | 9.42E+01 | 7.17E+00 | **5.26E-03** | 1.71E+02 | 2.98E+01 | 2.68E-02 | 3.23E+01 | 3.54E+00 |
| 15 | 4.08E-01 | 1.13E+02 | 6.84E+00 | 5.01E-01 | 1.63E+02 | 1.71E+01 | 2.70E-02 | 3.25E+01 | 4.24E+00 |
| 16 | -7.42E-03 | 2.57E+00 | 4.60E-01 | -1.15E-01 | 3.44E+01 | 1.25E+01 | **-1.37E-04** | 1.65E+00 | -3.73E-01 |
| 17 | -3.07E-02 | 8.64E+00 | 4.62E-01 | -2.21E-02 | 7.84E+00 | 4.39E-01 | **-2.31E-04** | 2.78E+00 | **4.32E-01** |
| 18 | **-6.16E-02** | 3.37E+02 | **4.84E+00** | -5.71E-01 | 3.58E+02 | 5.81E+01 | -3.22E-02 | 3.88E+02 | 2.74E+01 |
| 19 | -1.65E+01 | 4.28E+03 | 4.23E+02 | -8.23E+00 | 2.76E+03 | 6.70E+02 | **-5.82E-02** | 7.02E+02 | 4.58E+02 |
| 20 | -1.67E+01 | 3.97E+03 | 3.71E+02 | -8.54E+00 | 2.51E+03 | 7.96E+02 | **-8.34E-02** | 1.01E+01 | 4.21E+02 |
| 21 | 4.42E-01 | 2.04E+02 | 1.54E+01 | 5.05E-01 | 2.47E+02 | 1.46E+01 | 1.68E-02 | 2.02E+02 | 1.53E+01 |
| 22 | 4.40E-01 | 2.04E+02 | 5.13E+00 | 5.28E-01 | 2.54E+02 | 1.11E+01 | **1.68E-02** | 2.02E+02 | 5.08E+00 |
| 23 | 3.16E-01 | 1.47E+02 | 4.42E+01 | 5.61E-01 | 2.36E+02 | 2.93E+01 | **1.20E-02** | 1.45E+02 | 4.37E+01 |
| win | 15 | 17 | 14 | 15 | 14 | 17 | - | - | - |
| lose | 6 | 5 | 6 | 8 | 8 | 5 | - | - | - |
| draw | 2 | 1 | 3 | 0 | 1 | 0 | - | - | - |

mode known as lateral positioning in nature [16]. The optimization process of the MFO algorithm can be abstractly understood as two behaviors of the moth searching for flame and its abandoning flame [33]. Over iterations of the optimization process, moths and flames are manipulated with the simulated formula to update their positions. The moth is the actual search subject in the search space, and the flame is the best location so far. Therefore, if the moth finds a better solution, each moth searches for and updates the flame near the marked flame. The moth can get its optimal solution through this process. The set $Mo$ of the moth is expressed as follows:

$$Mo = \begin{bmatrix} Mo_{1,1} & Mo_{1,2} & \dots & Mo_{1,d} \\ Mo_{2,1} & Mo_{2,2} & \dots & Mo_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ Mo_{n,1} & Mo_{n,2} & \dots & Mo_{n,d} \end{bmatrix} \tag{1}$$

where $n$ is the number of moths, and $d$ is the number of variables (dimension). The fitness value $MF$ of the moth is expressed as follows:

$$\mathrm{MF} = \begin{bmatrix} MF_1 & MF_2 & \cdots MF_n \end{bmatrix}^T \tag{2}$$

Another critical element of the MFO algorithm is flame. The set of the flame is represented by matrix $FL$ as follows:

$$FL = \begin{bmatrix} FL_{1,1} & FL_{1,2} & \dots & FL_{1,d} \\ FL_{2,1} & FL_{2,2} & \dots & FL_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ FL_{n,1} & FL_{n,2} & \dots & FL_{n,d} \end{bmatrix} \tag{3}$$

In this matrix, $n$ is the number of moths, and $d$ is the number of variables (dimension). The matrix $FF$ represents the fitness value of the flame:

$$\mathrm{FF} = \begin{bmatrix} FF_1 & FF_2 & \cdots FF_n \end{bmatrix}^T \tag{4}$$

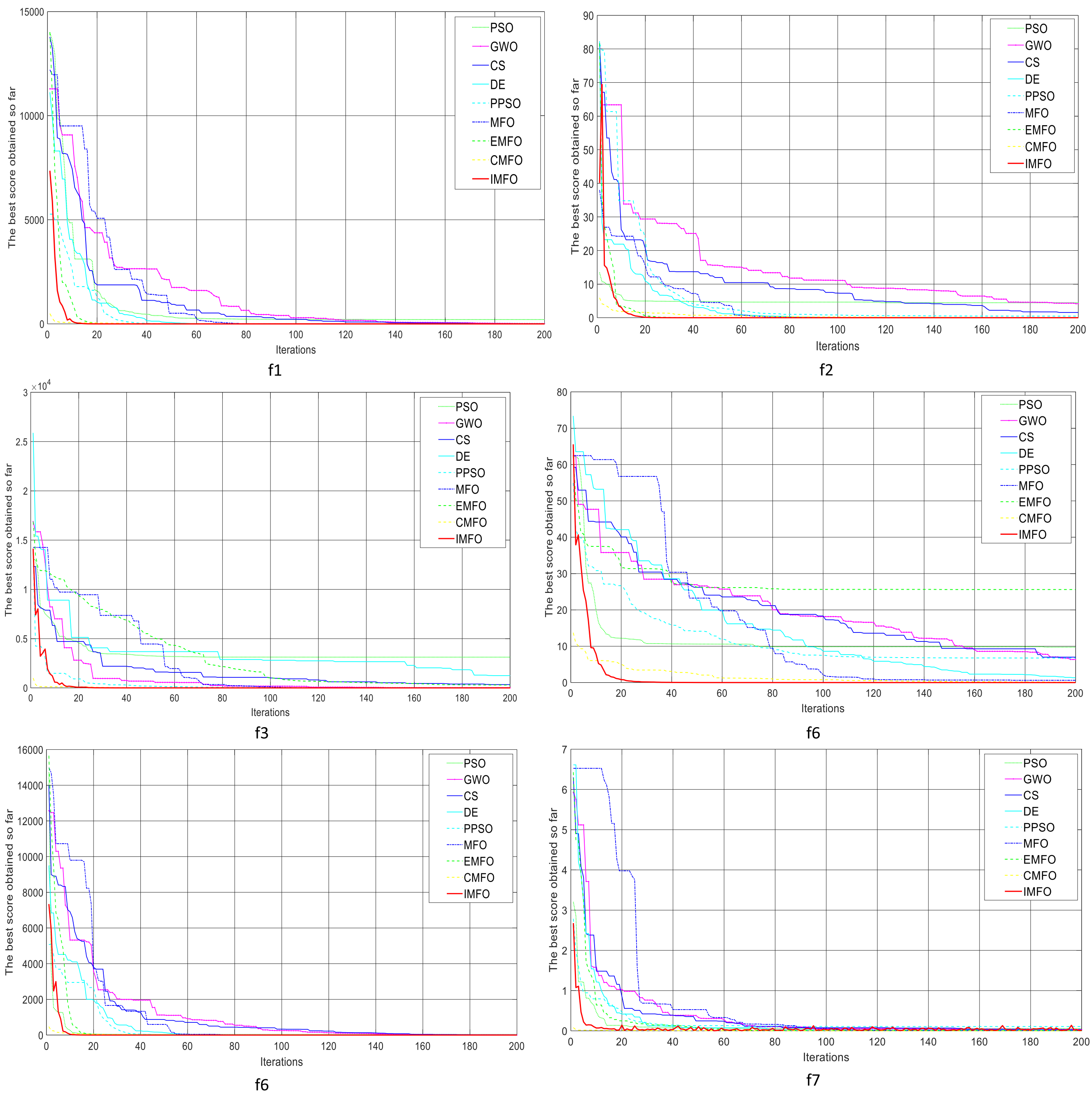


**FIGURE 2.** **Comparison of convergence curves of the best values the proposed IMFO with the other algorithms, such as the PSO [11], GWO [12], CS [13], DE [14], PPSO [15], MFO [16], EMFO [33], CMFO [34] for chosen functions: f1, f2, f3, f4, f6, and f7.**

The moth-flame optimization algorithm is approximately to find the global optimal triple

$$MFO = (I, P, T) \tag{5}$$

In the formula, $I$ is the function of random moth number and corresponding fitness value. The primary function $P$ makes the moth move in the solution space. It receives the matrix Mo and returns the updated Mo, which can be expressed as follows:

$$I : \emptyset \rightarrow \{Mo, MF\}, \quad P : Mo \rightarrow Mo \tag{6}$$

The $T$ function is a termination condition function. If the termination condition is met, the $T$ function returns true; otherwise, it returns false. The initial solution is generated by function $I$, and the objective function value is calculated. The $P$ function runs iteratively until the $T$ function returns true. Update the position of each moth relative to the flame using the following equation:

$$Mo_i = S(Mo_i, FL_j) \tag{7}$$

where $Mo_i$ is the ith moth, $FL_j$ is the jth flame, $S$ is a spiral function.

**TABLE 3.** Test results of the proposed IMFO, PPSO and PSO algorithms for selected 23 functions.

| Test Func. | PSO | | | PPSO | | | **IMFO** | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* |
| 1 | 9.25E-07 | 6.45E-04 | 1.61E-03 | 6.55E-11 | **2.79E-08** | 8.03E-08 | **2.82E-15** | 3.40E-08 | 1.42E-07 |
| 2 | 2.80E-03 | 1.99E+00 | 4.26E+00 | 4.52E-03 | 8.19E-01 | 3.64E+00 | **1.32E-04** | 1.58E-01 | **5.06E-03** |
| 3 | 6.10E-01 | 4.28E+02 | 7.40E+02 | 3.01E+01 | 5.68E-01 | 1.68E+03 | 1.14E-04 | 1.37E-01 | 9.20E-02 |
| 4 | 1.50E-03 | 1.05E-01 | 2.19E+00 | 5.70E-06 | **9.53E-06** | 3.15E-05 | 5.21E-13 | 6.27E-08 | **5.30E-05** |
| 5 | 1.54E-02 | 2.02E+01 | 8.80E-02 | 6.18E-02 | 2.02E+01 | 7.74E-02 | 1.67E-03 | 2.01E+01 | 8.83E-02 |
| 6 | 4.42E-03 | 4.28E+00 | 1.25E+00 | 1.63E-02 | 3.95E+00 | 7.65E-01 | **2.09E-04** | 2.51E+00 | **9.76E-01** |
| 7 | 2.07E-03 | 1.54E+00 | 1.67E+00 | **1.45E-03** | 2.52E+00 | 9.02E-01 | 1.70E-03 | 2.06E-01 | 1.53E-01 |
| 8 | 1.80E-02 | 1.54E+01 | 7.56E+00 | **2.45E-04** | 2.95E+00 | 1.86E+00 | 4.90E-04 | 5.90E+00 | 3.71E+00 |
| 9 | 2.35E-02 | 2.12E+01 | 7.74E+00 | 7.34E-02 | 1.72E+01 | 2.54E+00 | 8.44E-04 | 1.02E+01 | **8.25E-02** |
| 10 | 3.60E-02 | 3.13E+01 | 1.00E+01 | 8.20E-02 | 2.01E+01 | 6.05E+00 | **1.08E-03** | 1.30E+01 | 6.18E+00 |
| 11 | 5.94E-01 | 9.13E+02 | 3.12E+02 | 5.70E-02 | 5.32E+02 | 1.77E+02 | 8.78E-02 | 1.06E+03 | **3.52E+02** |
| 12 | 7.36E-01 | 1.10E+03 | 2.25E+02 | 3.29E+00 | 1.16E+03 | 1.67E+02 | 1.03E-01 | 1.25E+03 | 1.88E+02 |
| 13 | 4.25E-04 | **8.33E-01** | **2.08E-01** | 3.28E-03 | 1.11E+00 | 2.07E-01 | **9.46E-05** | 1.14E+00 | 2.29E-01 |
| 14 | 2.95E-02 | 3.60E+01 | 8.14E+00 | **3.46E-03** | 2.19E+01 | 1.91E+00 | 2.70E-03 | 3.27E+01 | 3.57E+00 |
| 15 | 3.00E-02 | 3.64E+01 | 9.14E+00 | 1.17E-01 | 3.58E+01 | 4.25E+00 | 2.73E-03 | 3.29E+01 | 4.29E+00 |
| 16 | -1.29E-03 | 1.69E+00 | 6.03E-01 | -1.13E-03 | 1.01E+00 | 2.49E-01 | **-1.39E-04** | 1.66E+00 | 3.77E-01 |
| 17 | -2.55E-03 | 3.11E+00 | 4.86E-01 | -1.01E-02 | 3.07E+00 | 3.76E-01 | -2.34E-04 | 2.81E+00 | 4.37E-01 |
| 18 | -2.86E-01 | 3.85E+02 | 4.48E+01 | -7.46E-01 | 3.18E+02 | 4.01E+01 | **-3.26E-02** | **3.92E+02** | **2.77E+01** |
| 19 | -7.34E-01 | 8.47E+02 | 3.67E+02 | **-2.98E-01** | 3.99E+02 | 2.44E+02 | -5.88E-02 | 7.09E+02 | 4.62E+02 |
| 20 | -9.42E-01 | 6.35E+02 | 3.83E+02 | -3.78E+00 | 6.30E+02 | 3.13E+02 | **-8.42E-02** | 1.02E+01 | 4.26E+02 |
| 21 | 1.65E-01 | 2.11E+02 | 9.84E+00 | 7.58E-01 | 2.27E+02 | 1.18E+01 | **1.69E-02** | 2.04E+02 | 1.54E+01 |
| 22 | 1.64E-01 | 2.10E+02 | 1.01E+01 | 5.23E-01 | 1.87E+02 | 9.16E+00 | **1.69E-02** | 2.04E+02 | 5.13E+00 |
| 23 | 1.70E-01 | 1.88E+02 | 5.76E+01 | **4.15E-01** | 1.48E+02 | 2.59E+01 | 1.21E-02 | 1.47E+02 | 4.42E+01 |
| win | 14 | 16 | 15 | 13 | 15 | 14 | - | - | - |
| lose | 6 | 5 | 8 | 8 | 7 | 8 | - | - | - |
| draw | 3 | 2 | 0 | 2 | 1 | 1 | - | - | - |

The helix should satisfy the following three conditions: the starting point is the moth, the endpoint is flame, and the floating range does not exceed the search space. Based on these three points, the logarithmic helix function of MFO algorithm is defined as follows:

$$S\left(Mo_i, FL_i\right) = D_i \bullet e^{bt} \bullet \cos\left(2\pi t\right) + FL_j \tag{8}$$

$$D_i = \left|FL_j - Mo_j\right| \tag{9}$$

where, $D_i$ is the distance between the $i^{th}$ moth and the $j^{th}$ flame; $b$ is a constant defining the logarithmic helix; $t$ is the random number between $[-1, 1]$, indicating the proximity between the next position of the moth and the flame (t = 1 is the closest to the flame, t = − 1 is the farthest from the flame). Indicates the closeness of the moth's next position to the flame (t = 1 is the closest to the flame, t = −1 is the farthest from the flame). In order to explore the solution space globally and ensure the fast convergence speed of the MFO algorithm, an adaptive updating mechanism of flame number is proposed to reduce the number of flames adaptively in the iteration process. The formula for updating flame number is as follows:

$$F^N = round(N - t\frac{N-1}{T}) \tag{10}$$

where, $N$ is the maximum number of flames; $t$ is the current number of iterations; $T$ is the maximum number of iterations.

### B. MINIMUM CROSS-ENTROPY FOR MULTI THRESHOLD

The multithreshold segmentation of color image plays a significant role in many areas of application, such as in the field of medicine, intelligent transportation, video monitoring, industrial production, and so on [24]. The threshold-based scheme is one of the powerful image segmentation algorithms by choosing a few thresholds to distinguish the target from the surrounding pixels [23]. For example, the bi-level

**TABLE 4.** Test results of the proposed IMFO, DE and EMFO algorithms for selected 23 functions.

| Test Func. | DE | | | EMFO | | | **IMFO** | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* |
| 1 | 1.85E-06 | 1.29E-03 | 3.21E-03 | 1.31E-10 | 2.19E-08 | **1.91E-08** | **2.85E-15** | 3.43E-08 | 1.43E-07 |
| 2 | 5.47E-03 | 3.83E+00 | 8.52E+00 | **8.90E-04** | 1.48E+00 | 7.27E+00 | 1.33E-04 | 1.60E-01 | **5.11E-03** |
| 3 | 1.22E+00 | 8.55E+02 | 1.48E+03 | 6.02E+01 | 1.00E+00 | 3.35E+03 | **1.15E-04** | 1.38E-01 | 9.29E-02 |
| 4 | 2.99E-03 | 2.09E-01 | 4.38E+00 | 1.14E-05 | 1.90E-05 | 1.00E-05 | **5.26E-13** | 6.33E-08 | **5.35E-05** |
| 5 | 2.91E-02 | 2.03E+01 | 8.76E-02 | 1.22E-01 | 2.04E+01 | 6.65E-02 | 1.69E-03 | 2.03E+01 | 8.92E-02 |
| 6 | 8.64E-03 | 6.04E+00 | 1.52E+00 | 3.23E-02 | 5.38E+00 | 5.54E-01 | 2.11E-04 | 2.54E+00 | 9.86E-01 |
| 7 | 4.12E-03 | 2.88E+00 | 3.19E+00 | 2.90E-02 | 4.84E+00 | 1.65E+00 | **1.72E-05** | 2.08E-01 | 1.55E-01 |
| 8 | 3.55E-02 | 2.49E+01 | 1.14E+01 | **8.08E-07** | 1.35E-04 | **1.05E-04** | 4.95E-04 | 5.96E+00 | 3.75E+00 |
| 9 | 4.61E-02 | 3.23E+01 | 1.54E+01 | 1.46E-01 | 2.43E+01 | 4.99E+00 | 8.53E-04 | 1.03E+01 | 8.33E-02 |
| 10 | 7.10E-02 | 4.96E+01 | 1.39E+01 | 1.63E-01 | 2.72E+01 | 5.93E+00 | **1.09E-03** | 1.31E+01 | 6.24E+00 |
| 11 | 1.10E+00 | 7.66E+02 | 2.71E+02 | 2.62E-02 | 4.37E+00 | 2.05E+00 | 8.87E-02 | 1.07E+03 | 3.56E+02 |
| 12 | 1.37E+00 | 9.57E+02 | 2.61E+02 | 6.47E+00 | 1.08E+03 | 1.46E+02 | 1.04E-01 | 1.26E+03 | 1.90E+02 |
| 13 | 7.55E-04 | 5.28E-01 | 1.87E-01 | 6.47E-03 | 1.08E+00 | 1.85E-01 | 9.56E-05 | 1.15E+00 | 2.31E-01 |
| 14 | 5.63E-02 | 3.94E+01 | 1.27E+01 | **6.65E-02** | 1.11E+01 | 2.40E-01 | 2.73E-03 | 3.30E+01 | 3.61E+00 |
| 15 | 5.72E-02 | 4.00E+01 | 1.40E+01 | 2.32E-01 | 3.87E+01 | 4.21E+00 | 2.76E-03 | 3.32E+01 | 4.33E+00 |
| 16 | -2.45E-03 | 1.72E+00 | 8.28E-01 | -2.13E-03 | 3.54E-01 | 1.21E-01 | **-1.40E-04** | 1.68E+00 | **3.81E-01** |
| 17 | -4.86E-03 | 3.40E+00 | 5.35E-01 | -1.99E-02 | 3.32E+00 | **3.15E-01** | -2.36E-04 | 2.84E+00 | 4.41E-01 |
| 18 | **-5.39E-03** | 3.77E+02 | 6.19E+01 | -1.46E+00 | 2.44E+02 | 5.24E+01 | -3.29E-02 | 3.96E+02 | 2.80E+01 |
| 19 | -1.41E+00 | 9.85E+02 | 2.71E+02 | -5.37E-01 | 8.95E+01 | 2.58E+01 | -5.94E-02 | 7.16E+02 | 4.67E+02 |
| 20 | -1.80E+00 | 1.26E+03 | **2.41E+02** | -7.48E+00 | 1.25E+03 | 2.00E+02 | **-8.51E-02** | 1.03E+01 | 4.30E+02 |
| 21 | 3.13E-01 | 2.19E+02 | 4.24E+00 | 1.50E+00 | 2.50E+02 | 8.17E+00 | **1.71E-02** | 2.06E+02 | 1.56E+01 |
| 22 | 3.11E-01 | 2.17E+02 | 1.51E+01 | 1.03E+00 | 1.71E+02 | 1.32E+01 | **1.71E-02** | 2.06E+02 | 5.18E+00 |
| 23 | 3.28E-01 | 2.30E+02 | 7.11E+01 | 8.98E-01 | 1.50E+02 | 7.66E+00 | 1.22E-02 | 1.48E+02 | 4.46E+01 |
| **win** | 11 | 13 | 12 | 10 | 8 | 7 | - | - | - |
| lose | 7 | 8 | 10 | 9 | 7 | 8 | - | - | - |
| draw | 3 | 2 | 1 | 4 | 7 | 7 | - | - | - |

thresholding problem just has to select one limit to classify objects and surroundings into two classes, which is quickly applied. However, multilevel thresholding is more prevalent in problem-solving activities such as mixed-type text interpretation and color image segmentation [40]. Threshold processing is a practical tool for the segmentation of images. Single threshold image segmentation can only be considered an actual design problem, whereas multi thresholds can be viewed as a multi-objective programming optimization problem [28]. Multi threshold image segmentation is to use the threshold vector T = $\{t_1, t_2, \ldots, t_K\}$ composed of $K$ thresholds to divide the image I into $K+1$ region $\{C_0, C_1, \ldots, C_K\}$. The minimum cross entropy segmentation method is used to select the global optimal threshold vector $T^*$. The optimal threshold $T^*$ needs to be satisfied as the following expression.

$$T^* = argmin\{f\ (t_1, \ldots, t_K)\} \tag{11}$$

where f is the objective function for the multiple thresholds for image segmentation. Finding the target vector with the minimum fitness value $T^*$ is the best threshold vector obtained from the image. The implementation process of the multi-threshold image segmentation can be regarded as a single objective optimization problem. The optimization process combined with the swarm intelligence optimization algorithm is actually the optimal threshold combination to solve the fitness function of minimum cross-entropy multi-threshold segmentation.

**TABLE 5. Test results of the proposed IMFO, CS and CMFO algorithms for selected 23 functions.**

| Test Func. | CS | | | CMFO | | | IMFO | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* | *Best* | *Mean* | *Std.* |
| 1 | 5.64E-10 | 1.34E-09 | 1.16E-10 | 5.96E-12 | 1.75E-09 | **6.81E-10** | **2.85E-12** | 3.43E-11 | 1.43E-07 |
| 2 | 1.15E-02 | 2.75E+00 | 2.63E+00 | **1.67E-02** | 4.91E+00 | 5.99E+00 | **1.33E-04** | 1.60E-01 | **5.11E-04** |
| 3 | 7.01E-04 | 1.67E-01 | 3.27E-01 | 5.26E-04 | 1.55E-01 | 2.43E-01 | **1.15E-04** | 1.38E-01 | **9.29E-02** |
| 4 | 5.15E-14 | 1.23E-13 | **3.09E-14** | 1.75E-14 | 5.16E-02 | 3.68E-02 | **5.26E-16** | 6.33E-09 | 5.35E-04 |
| 5 | 8.80E-02 | 2.10E+01 | 4.57E-02 | 7.12E-02 | 2.09E+01 | 4.32E-02 | 1.69E-03 | 2.03E+01 | **8.92E-02** |
| 6 | 1.68E-01 | 4.01E+01 | 1.01E+00 | 1.20E-01 | 3.53E+01 | 3.17E+00 | **2.11E-04** | 2.54E+00 | 9.86E-01 |
| 7 | 1.12E-03 | 2.66E-01 | 9.69E-02 | 3.98E-01 | 1.17E+02 | 5.53E+01 | 1.72E-05 | 2.08E-01 | **1.55E-01** |
| 8 | 5.36E-02 | 1.28E+01 | 5.03E+00 | 9.20E-01 | 2.71E+02 | 5.57E+01 | **4.95E-04** | 5.96E+00 | 3.75E+00 |
| 9 | 6.78E-01 | 1.61E+02 | 8.98E+00 | 9.12E-01 | 2.68E+02 | 7.98E+01 | **8.53E-04** | 1.03E+01 | 8.33E-02 |
| 10 | 6.69E-01 | 1.59E+02 | 8.74E+00 | **1.14E-02** | 3.36E+02 | 6.37E+01 | 1.09E-03 | 1.31E+01 | 6.24E+00 |
| 11 | 3.24E+01 | 7.73E+03 | 3.18E+02 | 1.41E+01 | 4.14E+03 | 7.12E+02 | **8.87E-02** | 1.07E+03 | 3.56E+02 |
| 12 | **3.31E-03** | 7.87E-01 | 3.56E-01 | 1.47E-03 | 4.31E-01 | 1.09E-01 | 1.04E-01 | 1.26E-01 | **1.90E-01** |
| 13 | 3.64E-06 | **8.67E-04** | 8.12E-04 | 7.65E-03 | 2.25E+00 | 4.19E-01 | **9.56E-05** | 1.15E+00 | 2.31E-01 |
| 14 | 6.54E-01 | 1.56E+02 | 1.08E+01 | 1.05E+00 | 3.09E+02 | 5.61E+01 | 2.73E-03 | 3.30E+01 | 3.61E+00 |
| 15 | 8.14E-01 | 1.94E+02 | 9.44E+00 | 9.99E-01 | 2.94E+02 | 2.99E+01 | **2.76E-03** | 3.32E+01 | 4.33E+00 |
| 16 | -1.47E-02 | 3.49E+00 | 5.46E-01 | -2.29E-01 | 6.72E+01 | 2.46E+01 | -1.40E-04 | 1.68E+00 | 3.81E-01 |
| 17 | -6.11E-02 | 1.45E+01 | 4.92E-01 | -4.40E-02 | 1.29E+01 | 4.46E-01 | **-2.36E-04** | 2.84E+00 | 4.41E-01 |
| 18 | -1.20E+00 | 2.85E+02 | 6.94E+01 | **-1.11E-03** | 3.27E+02 | 8.87E+01 | -3.29E-02 | 3.96E+02 | 2.80E+01 |
| 19 | **-3.83E-02** | **1.79E+00** | 3.89E+02 | -1.64E+01 | **4.82E+00** | **8.83E+00** | -5.94E-02 | 7.16E+02 | 4.67E+02 |
| 20 | -3.33E+01 | 7.93E+03 | 3.20E+02 | -1.70E+01 | 5.00E+03 | 1.17E+03 | **-8.51E-02** | 1.03E+01 | 4.30E+02 |
| 21 | 8.67E-01 | 2.06E+02 | 1.56E+01 | 9.94E-01 | 2.92E+02 | 1.40E+01 | 1.71E-02 | 2.06E+02 | 1.56E+01 |
| 22 | **8.64E-01** | 2.06E+02 | 5.18E+00 | **1.04E-03** | 3.07E+02 | 1.71E+01 | 2.71E-02 | 2.06E+02 | 5.18E+00 |
| 23 | 6.20E-01 | 1.48E+02 | 4.46E+01 | 1.11E+00 | 3.27E+02 | 1.48E+01 | **1.22E-02** | 1.48E+02 | **4.46E+01** |
| **win** | 13 | 15 | 13 | 10 | 11 | 10 | - | - | - |
| lose | 7 | 6 | 8 | 9 | 7 | 8 | - | - | - |
| draw | 3 | 2 | 2 | 4 | 5 | 5 | - | - | - |

## III. ENHANCING MOTH-FLAME ALGORITHM

This section presents the proposed improving MFO algorithm (IMFO) by modifying with three factors, i.e., hybridizing with the Lévy flight, adapting weight, and descending curvilinear strategy. The presentation is split into two subsections: the proposed part and the experimental results and discussion part in detail as follows.

### A. IMPROVING MOTH-FLAME OPTIMIZATION ALGORITHM

In order to improve MFO algorithm (IMFO), the updating formula of flame and moth is modified with three factors, i.e., hybridizing with the Lévy flight, adapting weight, and descending curvilinear strategy. The suggested scheme of enhancing the optimization algorithm is described details as beginning with hybridizing with the Lévy flight as follows.

Lévy flight is introduced here as a kind of random walk, which is characterized by a large number of short-distance walks and a small number of long-distance jumps. It can be used to simulate random or pseudo-random natural phenomena. Since it is difficult to implement the original formula, we generally use the improved planning mode, which is expressed as follows.

$$\text{Levy}(\beta) \sim \frac{\mu}{|\gamma|^{1/\beta}} \tag{12}$$

Lévy can be computed by components of Lévy flight; $\mu \sim N(0,\sigma^2)$; $\gamma \sim N(0, 1)$.

$$\sigma = \left\{ \frac{\Gamma(1+\beta) sin(\pi\beta/2)}{\beta\Gamma\left[(1+\beta)/2\right] 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}} \tag{13}$$

The application of Lévy flight to the MFO algorithm is implemented through updating the position of moths relative to the

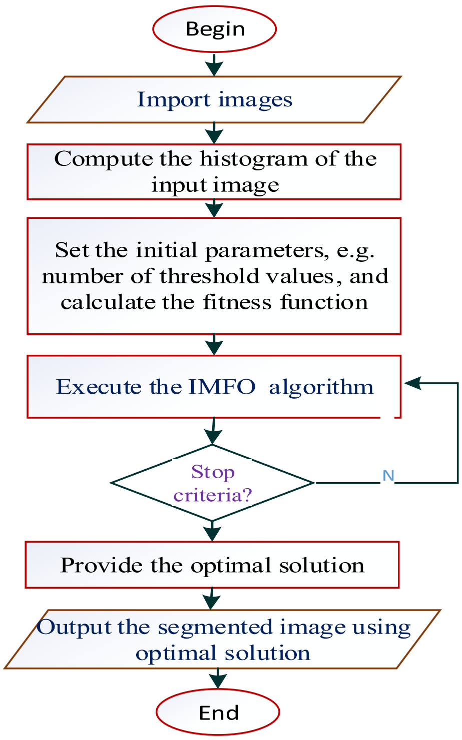


**FIGURE 3.** **Flow chart of multi threshold image segmentation using the IMFO.**

flame in Eq.(7) is figured out as follows.

$$Mo_i = Mo_i \circledast \text{Levy}(\beta) \tag{14}$$

where, $\circledast$ is the point multiplication operation. Additionally, because of the MFO algorithm, the updating mechanism of the moth position is realized by logarithmic helix function. Still, this function only defines the moth flying to the flame, which makes the moth fall into the local optimum easily while dealing with complicated problems, and there are some deficiencies in the global optimization. The adaptive weight method used in this article reduces the value of the adaptive weight when the moth looks for the optimal solution near the flame, which can enhance the searching ability of the moth near the optimal solution and improve the local optimization ability of the moth. Thus, the logarithmic helix function of the MFO algorithm used in Eq.(8) also can be enhanced for the speed converge in optimization processing by adding the inertia weight adaption.

The specific calculation formula of adaptive weight $w$ is as follows:

$$w = 1 + \sin(\pi + \frac{\pi t}{2T}) \tag{15}$$

Therefore, the original formula of position Eq.(8) is changed by modifying as follows.

$$S\,(Mo_i, FL_i) = D_i \bullet e^{bt} \bullet \cos\,(2\pi t) + wFL_j \tag{16}$$

The adaptive weight $w$ is multiplied by the jth flame $FL_j$, when the moth approaches the flame, the value of $w$ will decrease, which improves the moth's local optimization ability and avoids the moth missing the optimal solution.

Moreover, because there are a large number of random states in the swarm behavior of moths, it needs to be tested repeatedly, which leads to the time-consuming of the algorithm. The updating mechanism of flame number is changed from linear descent to curvilinear descent, which improves the convergence speed of the adaptive flame number and the convergence speed of the algorithm. The updated formula of the number of adaptive flames $F^N$ is as follows.

$$F^N = round(\frac{T}{t + (T/N)}) \tag{17}$$

where, $N$ is the maximum number of flames; $t$ is the current number of iterations; $T$ is the maximum number of iterations. Algorithm 1 lists the pseudo-code of the proposed IMFO.

**Algorithm 1** Pseudo-Code of the IMFO

```
Initialization:
Initialize the position of moths and the parameters
Iteration:
while iteration<max_iteration do
Update the flame by Eq. (10)
Update inertia weight coefficient by Eq. (15)
MF=Fitness function(Mo)
if iteration=1
    Sor t the first population of moths and tag the best position by flames
else
    Sort the other population of moths and update the other flames
end if
    for i = 1 : n do
        for j = 1 : d do
            Update b and t as Eq.(8)
            Update the distance d with the Eq. (9)
            According to the conditions, the position of the moth updated by Eq. (16)
        end for
    end for
    Update the position of the current search agent with the Eq. (14)
    iteration=iteration+1
end while
Output:
Global optimal value and global optimal solution
```

The detailed steps of the IMFO optimization algorithm are summarized based on the above discussed as follows.

Step 1: Set the initialization space, generate population n for moths randomly, the maximum number of iterations, calculate the fitness function, tag the best locations by flames, and S set to a spiral function.

Step 2: Update the parameter b and t using Eq. (8); Calculate D for the corresponding moths Eq.(9); Sort and assign flame Eq.(2). Update the moth Eq.(7).


**FIGURE 4.** Selected original images.

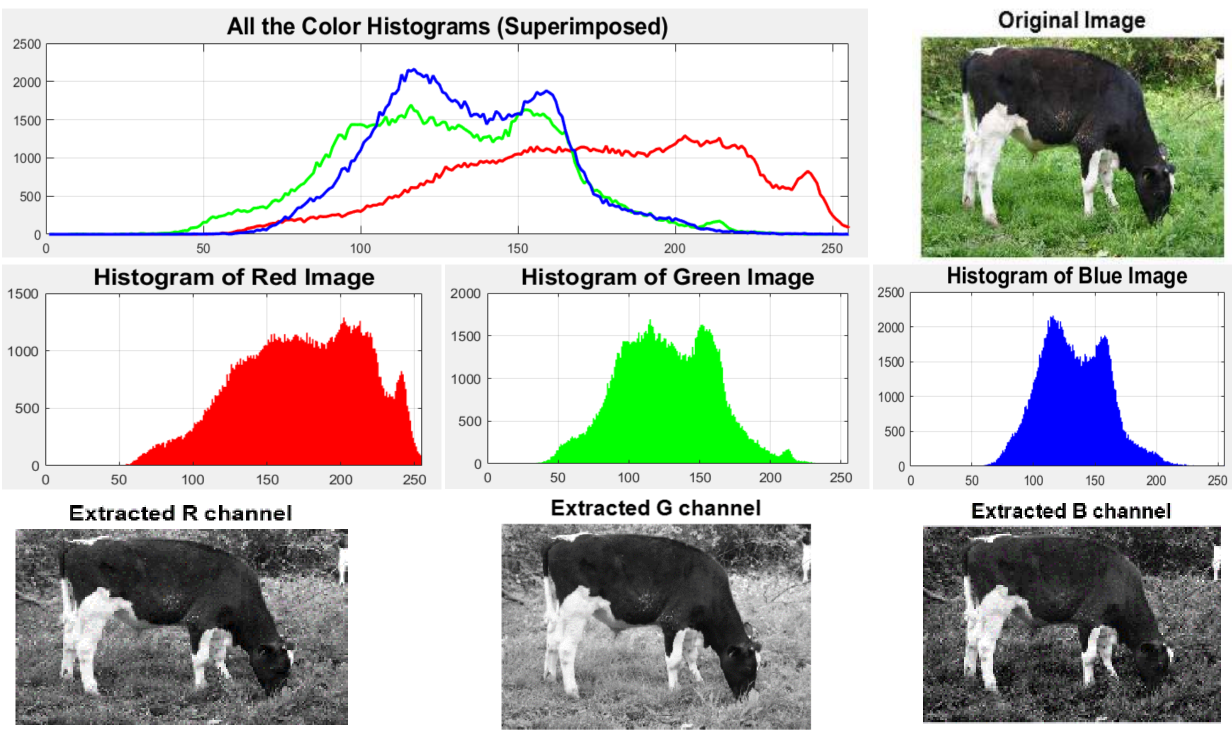


**FIGURE 5.** Visually obtained three channels (RGB) of the proposed IMFO for the image 06.

Step 3: Calculate adaptive weight Eq.(15); Update the S function Eq.(16); Compute the components of Lévy flight: $\mu$, $\sigma$, and $\gamma$ in Eq (13) for calculation of Eq.(12); Update the new moths with Eq.(14).

Step 4: If the number of iterations reaches the maximum or the optimal position meets the convergence conditions, the algorithm is terminated, and the search result is the output: the globally optimal moth position and

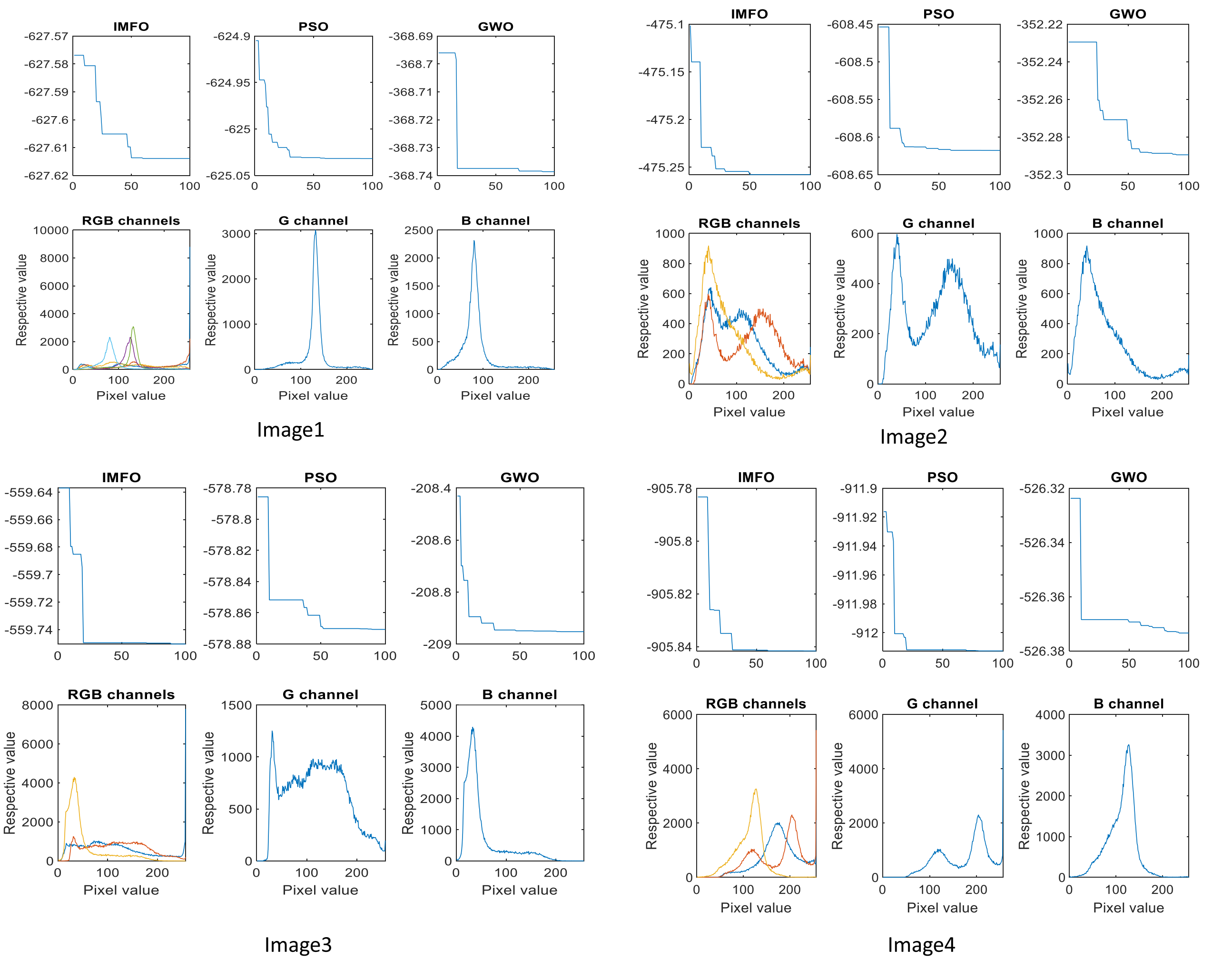


**FIGURE 6. Comparison of the suggested IMFO scheme with the methods of the PSO [28], and GWO [40] and the three channels (RGB) visually derived for images 01 to 04 with thresholds set at 8.**

its corresponding fitness function value. Otherwise, return to Step 2.

Figure 1 shows the flowchart of the proposed IMFO algorithm.

### B. EXPERIMENTAL RESULTS AND DISCUSSION

In order to assess the possible well efficiency of the proposed IMFO algorithm, we use the chosen samples from the test suite CEC2013 [41] for certain benchmark functions. The reasons for only selecting twenty-three functions from CEC2013, 14 [41], [42] is because of the test suit have varieties functions of categories, e.g., the unimodal, multimodal, fixed-dimension modal, and composite benchmark functions. We selected several functions from each of the classes for presenting features in the guarantee of varieties of benchmark functions. The selected twenty-three functions are to give easily and fitly in a page layout presentation. Literatures [41], [43] give complete access to the test suites at CEC2013 that consists of some types of test models (A, B, C, and D as listed in Table1) for real number optimization functions. The selected functions include four unimodal test functions (f1-f4), six multi-modal test functions (f5-f10), ten fixed-dimension multimodal benchmark functions (f11-f20) and three sample composition functions (f21-f23).

The experimental results obtained from the proposed IMFO algorithm are compared in literature with various algorithms, e.g., the PSO [11], PPSO [15], CS [13], DE [14], MFO [16], GWO [12], EMFO [33], CMFO [34] algorithms. In order to allow the comparison of different algorithms, all values for the fitness function are reduced to the same minimum. Which one of the intelligent algorithms, competitive algorithm can typically consider a lower, optimal global cost, it would be the better one.

Table 1 lists the initialization range of boundary space, dimension, and description test benchmark functions.

Parameter settings for the optimization algorithms randomly initialize the number of solutions to 80, and each solution is set to Dim., of dimensions, and the search space range (for example, of aspect is −100 to 100) range as (Dim., and range listed in Table 1).

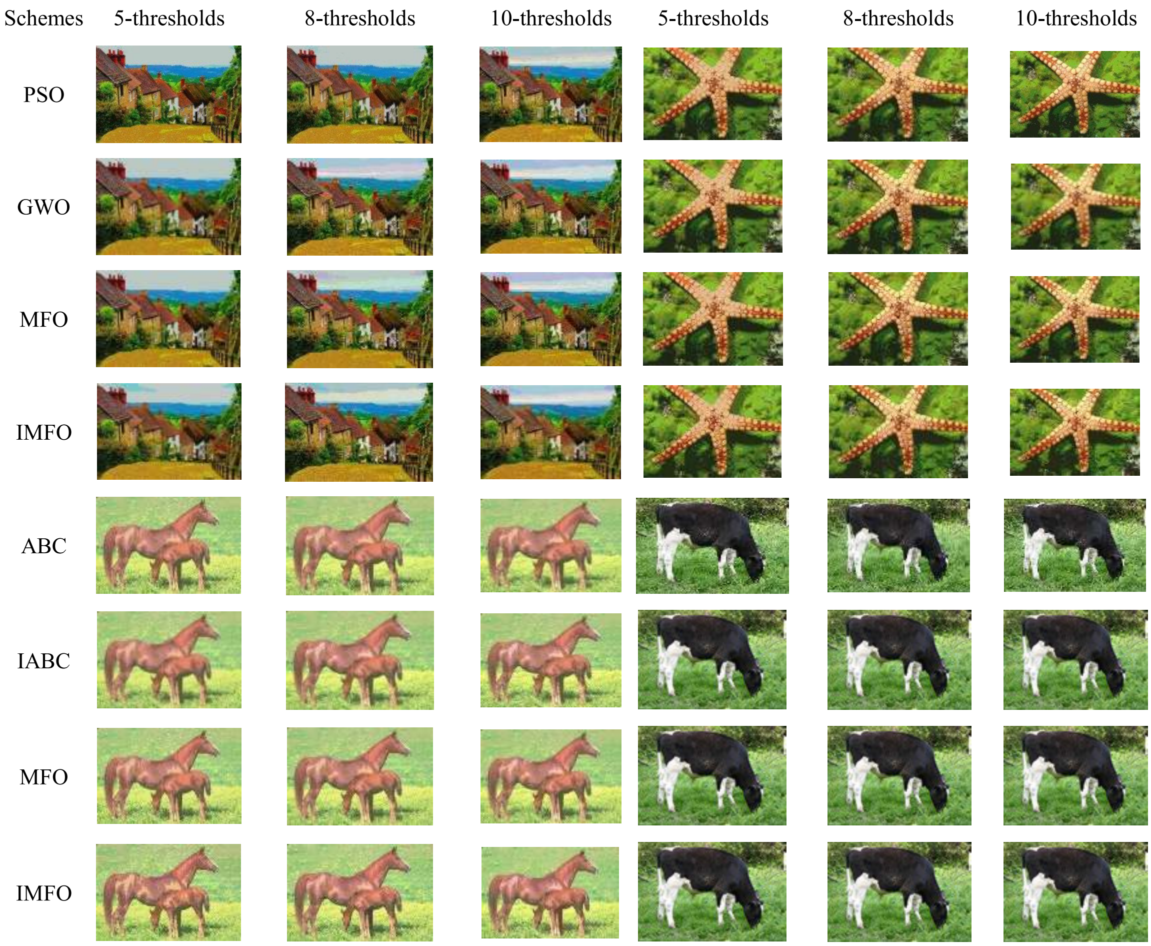


**FIGURE 7. Comparison of graphs obtained under different threshold values.**

The maximum number of iterations is 200. For a fair comparison, each algorithm is run 21 times. The parameter $b$ is set 1 for the IMFO, EMFO, CMFO, and original MFO algorithms. The variable $a$ is set 2 to 0 for the GWO algorithm. The weight $w$ is placed 0.9 to 0.4, and parameter$c_1$, and $c_2$ are set to 1.49455 for the PSO and PPSO algorithms. $\alpha$ is set to a range of 0.08 to 0,13, $\beta$ is set to 0.99, and $\gamma$ is set to a variety of 3 to 5 for the CS algorithm. F is set to a range of 0.6 to 0.9, and $cr$ is set to 0.2 for the DE algorithm.

Some experimental and comparative experiments are implemented to evaluate the proposed IMFO performance that is presented in detailed experimental data in Tables 2-5, and Figure 2.

In order to comprehensively assess the performance of the proposed IMFO algorithm, we recorded the best value, mean, and standard deviation of the optimization functions for the algorithm run 21 times. The smaller the value obtained by the algorithm, the better the optimization results.

The data tables about obtained results of the algorithms are the statistical obtained results, e.g., the optimal best value, average values, and standard deviation of the optimal values for test functions are compared algorithms to reflect the performance of the proposed algorithm. The consumed times of running optimization algorithms for the test functions are also considered to generate a statistic. The summarized symbols are "win", "lose", and "draw" that presented means as the comparison between the proposed IMFO with the other algorithms in paired for the best, lose, or draw. For example, if the best value of the obtained results of the IMFO for test function 1 is smaller (minimum function) than the PSO algorithm, the "win" is counted for adding 1, otherwise, the symbol of "lose" is counted one or the "draw" is counted one if it equals.

Observed from Tables 2 to 5, the results show that the proposed IMFO has a number of the "wins" that are more than the PSO, MFO, DE, CS, WGO, and PPSO. However, the figures for the IMFO are not more significant than the EMFO and CMFO algorithms much.

The execution consumed times of the proposed algorithm running for the test functions are also considered to generate a statistic. As added multiple the moth position vector with the Lévy flight operator, the time running is also affected that caused the executing test functions, especially for the unimodal, and fixed modal benchmark functions have the time consumption executing longer than the previous one. However, for the complicated test functions like the

**TABLE 6.** The metric of evaluation parameters used to measure segmented image results.

| | Parameters | Formulas | Remarks |
|---|---|---|---|
| 1. | *MSE* *and* *PSNR* | $MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[I(x,y) - I'(x,y)]^2$ ; $PSNR = 20\log_{10}\frac{255}{\sqrt{MSE}}$ | $MSE$ is a metric of the deviation, between actual and expected values. $PSNR$ is the ratio between maximum signal and the noise obtained from $MSE$. |
| 2. | *SSIM* | $SSIM(xx \in I, yy \in I') = \frac{(2\mu_{xx}\mu_{yy} + c_1) \times (2\sigma_{xx,yy} + c_2)}{(\mu_{xx}^2 x + \mu_{vv}^2 y + c_1) \times (\sigma_x^2 + \sigma_{vv}^2 + c_2)}$ | $SSIM$ *is* a parameter of structural similarity between the original image and the segmented image. where: $\mu_{xx}$ and $\mu_y$ are the average gray levels of the original image and the segmented image respectively; $\sigma_x^2$ and $\sigma_y^2$ are the variances of the two images respectively; $\sigma_{xy}$ is the covariance of the original image and the segmented image; $c_1$ and $c_2$ are constants. |
| 3 | FSIM | $\mathrm{FSIM} = \frac{\sum_{x\in\Omega} S_L(x) \times PC_m(x)}{\sum_{x\in\Omega} PC_m(x)}$ | $FSIM$ is a parameter of the feature similarity that is an index to evaluate image quality based on phase consistency and spatial gradient feature of the image. Where: $\Omega$ is the whole spatial domain of the image; $S_L(x)$ is used to evaluate the image similarity; $PC_m(x)$ is the phase similarity. |
| 4 | *EPI* | $EPI = \frac{\sum_{i=1}^{m}\lvert G_{R1} - G_{R2}\rvert}{\sum_{i=1}^{m}\lvert G_{R1} - G_{R2}\rvert}$ | $EPI$ is an Edge retention coefficient that represents the ability of the filter to preserve the horizontal or vertical edges of the image after processing. The higher the value, the stronger the retention ability. The numerator part represents the parameters after filtering, and the denominator represents the parameters before filtering; where $m$ is the number of image pixels; $G_{R1}$ and $G_{R2}$ are the gray values of left and right or upper and lower adjacent pixels respectively. |

**TABLE 7.** Comparison of the obtained results of the optimization of the proposed IMFO scheme with the PSO, MFO, GWO, ABC, IABC schemes for multilevel image segmentation based on a metric of the PSNR.

| Image | threshold | *PSNR* | | | | | |
|---|---|---|---|---|---|---|---|
| | | PSO | GWO | MFO | ABC | IABC | **IMFO** |
| Image1 | 5 | 28.90297 | 28.88233 | 28.88295 | **29.03561** | 28.87216 | 28.93651 |
| | 8 | 30.88260 | 30.76298 | 30.74016 | 30.63679 | 30.79367 | **30.90804** |
| | 10 | 31.58761 | 31.62937 | 31.34181 | 31.09397 | 31.59563 | **31.66930** |
| | 12 | 31.87878 | 31.99239 | 31.80178 | 31.49900 | 31.94559 | **32.00227** |
| Image2 | 5 | 29.57887 | 29.57887 | **29.58762** | 29.23925 | 29.57850 | 29.56536 |
| | 8 | 31.68925 | 31.86244 | 31.77791 | 30.98924 | **31.91567** | 31.87192 |
| | 10 | 32.36551 | 32.56902 | 32.39916 | 31.52746 | 32.51123 | **32.61501** |
| | 12 | 32.76274 | 32.97623 | 32.83233 | 32.16131 | 32.78393 | **33.00080** |
| Image3 | 5 | 30.87890 | 30.8789 | 30.87866 | 30.49437 | 30.88013 | **30.88640** |
| | 8 | 33.45137 | 33.50686 | 33.48056 | 32.82658 | 33.51200 | **33.59921** |
| | 10 | 34.20608 | **34.50588** | 34.02068 | 33.36215 | 34.38188 | 34.45018 |
| | 12 | 34.75214 | 34.99056 | 34.72816 | 33.64132 | 34.80053 | **35.03611** |
| Image4 | 5 | **31.29500** | 31.29015 | 31.29174 | 30.97254 | 31.29071 | 31.29185 |
| | 8 | **32.89962** | 32.89555 | 32.87188 | 32.22659 | 32.88825 | 32.86529 |
| | 10 | 33.19552 | **33.38643** | 33.20282 | 32.53222 | 33.28118 | 33.34261 |
| | 12 | 33.47538 | 33.54890 | 33.42869 | 32.69779 | 33.54831 | **33.54948** |
| Image5 | 5 | 31.46847 | 31.46843 | 31.43929 | 31.11899 | **31.47216** | 31.47114 |
| | 8 | 34.83435 | 34.88032 | 34.87776 | 33.57890 | 34.80603 | **34.91311** |
| | 10 | 36.02319 | 36.38910 | 35.84885 | 34.48454 | 36.16735 | **36.56246** |
| | 12 | 36.71580 | 37.35114 | 36.94982 | 35.61768 | 36.91715 | **37.54162** |
| Image6 | 5 | 31.02476 | 31.01885 | 31.02031 | 30.75351 | 31.02692 | **31.04430** |
| | 8 | 33.65560 | 33.71741 | 33.59673 | 32.78215 | 33.74083 | **33.79716** |
| | 10 | 34.55400 | 34.69403 | 34.22598 | 33.54275 | 34.74308 | **34.81645** |
| | 12 | 35.22347 | 35.36996 | 35.07352 | 34.33831 | 35.25926 | **35.56532** |

category of the composite and multimodal functions, the time running use of the proposed IMFO is as long as the other algorithms such as the EMFO, CMFO, and shorter than PPSO.

**TABLE 8.** Comparison of the obtained results of the optimization of the proposed IMFO scheme with the PSO, MFO, GWO, ABC, IABC schemes for multilevel image segmentation based on a metric of the SSIM.

| Image | threshold | *SSIM* | | | | | |
|---|---|---|---|---|---|---|---|
| | | PSO | GWO | MFO | ABC | IABC | **IMFO** |
| Image1 | 5 | 0.87378 | 0.87326 | 0.87328 | **0.87801** | 0.87305 | 0.87478 |
| | 8 | 0.90936 | 0.90720 | 0.90717 | 0.90611 | 0.90783 | **0.90985** |
| | 10 | 0.91904 | 0.92044 | 0.91713 | 0.91235 | 0.91944 | **0.92093** |
| | 12 | 0.92319 | **0.92486** | 0.92309 | 0.91741 | 0.92357 | 0.92478 |
| Image2 | 5 | 0.89382 | 0.89382 | 0.89379 | 0.88813 | 0.89383 | **0.89393** |
| | 8 | **0.91991** | 0.91959 | 0.91971 | 0.91310 | 0.91978 | 0.91956 |
| | 10 | 0.92695 | 0.92948 | 0.92693 | 0.91712 | 0.92890 | **0.93034** |
| | 12 | 0.93247 | 0.93593 | 0.93377 | 0.92564 | 0.93193 | **0.93604** |
| Image3 | 5 | 0.88034 | 0.88034 | 0.88049 | 0.87528 | 0.88036 | **0.88071** |
| | 8 | 0.92033 | 0.92126 | 0.92100 | 0.91163 | 0.92105 | **0.92179** |
| | 10 | 0.92964 | **0.93401** | 0.92971 | 0.91831 | 0.93225 | 0.93337 |
| | 12 | 0.93630 | **0.93970** | 0.93845 | 0.92382 | 0.93679 | 0.93967 |
| Image4 | 5 | **0.88811** | 0.88797 | 0.88803 | 0.88277 | 0.88797 | 0.88800 |
| | 8 | **0.92353** | 0.92339 | 0.92303 | 0.91016 | 0.92322 | 0.92296 |
| | 10 | 0.92868 | **0.93192** | 0.92871 | 0.91706 | 0.93001 | 0.93105 |
| | 12 | 0.93351 | 0.93424 | 0.93252 | 0.91895 | **0.93463** | 0.93445 |
| Image5 | 5 | **0.86320** | 0.86288 | 0.86267 | 0.85887 | 0.86289 | 0.86285 |
| | 8 | 0.90469 | **0.90524** | 0.90451 | 0.89324 | 0.90477 | 0.90489 |
| | 10 | 0.91882 | 0.92017 | 0.91677 | 0.90540 | 0.91967 | **0.92175** |
| | 12 | 0.92769 | 0.93174 | 0.92972 | 0.91700 | 0.92862 | **0.93396** |
| Image6 | 5 | 0.90214 | 0.90223 | 0.90179 | 0.89707 | 0.90237 | **0.90261** |
| | 8 | 0.93465 | 0.93576 | 0.93537 | 0.92547 | 0.93574 | **0.93614** |
| | 10 | 0.94362 | 0.94522 | 0.94250 | 0.93440 | 0.94496 | **0.94583** |
| | 12 | 0.94925 | 0.95107 | 0.94974 | 0.94255 | 0.95022 | **0.95234** |

Figure 2 shows the comparison of convergence curves of the best values the proposed IMFO with the other algorithms, such as the PSO, MFO, GWO, CS, DE, PPSO, EMFO, and CMFO for chosen functions: f1, f2, f3, f4, f6, and f7. It can be seen that the proposed IMFO produces faster convergence than the other algorithms in comparative experiments for the selected test functions.

## IV. THE IMFO FOR MULTITHRESHOLD IMAGE SEGMENTATION

The traditional multithreshold segmentation algorithms faced the computation problem; complicated time increases rapidly with whenever the number of thresholds increased. The intelligent optimization algorithm must be combined with the optimal threshold vector. In order to solve this problem well, the improved MFO algorithm is applied to the multithreshold image segmentation, where minimal cross-entropy is taken as the objective function for optimization.

### A. MULTITHRESHOLD IMAGE SEGMENTATION USING IMFO

Assume that $n$ thresholds for an original image can be divided into the various groups *T(n+1)*. Let $t_1, t_2, \ldots, t_n$.be $n$ thresholds of the image regions with $class_1 \in \{0, \ldots, t_1\}$, $class_2 \in \{t_1, \ldots, t_2\}$, …, $class_{n+1} \in \{t_n, \ldots, L\}$. The objective function [28] for the optimal $n$ thresholds is formulated as follows.

$$\left\{t_1^*, \ldots, t_n^*\right\} = argmin\left\{f\left(t_1, \ldots, t_n\right)\right\} \quad Subject\ to\ 0 < t_1 < t_2 < \ldots < t_n < L \tag{18}$$

where $\left\{t_1^*, \ldots, t_n^*\right\}$ are optimal obtained results from $n$ thresholds; $f\left(t_1, \ldots, t_n\right)$ is the objective function for the optimization of the multilevel image segmentation with optimum threshold values. The minimum cross-entropy scheme is used to determine the appropriate thresholds for image segmentation. The cross-entropy measures the theoretical distance of information for two on the same set of probability distributions [44]. Let $P\{p_1, p_2, \ldots, p_n\}$, and $Q\{q_1, q_2, \ldots, q_n\}$ be two probabilistic distributions. The cross-entropy can be expressed for two distributions of the $P$ and $Q$ as follows.

$$\mathrm{D}\left(P, Q\right) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{19}$$

The threshold values can be calculated based on optimizing the cross-entropy between the threshold version and the original image. The feature extraction of the image $I_s$ is figured out as follows.

$$I_s\left(x, y\right) = \begin{cases} u\left(1, t\right), & I\left(x, y\right) < t \\ u\left(t, L+1\right), & I\left(x, y\right) \geq t \end{cases} \tag{20}$$

**TABLE 9.** Comparison of the obtained results of the optimization of the proposed imfo scheme with the PSO, MFO, GWO, ABC, IABC schemes for multilevel image segmentation based on a metric of the FSIM.

| Image | threshold | FSIM | | | | | |
|---|---|---|---|---|---|---|---|
| | | PSO | GWO | MFO | ABC | IABC | **IMFO** |
| Image1 | 5 | 0.92547 | 0.92538 | 0.92515 | **0.92614** | 0.92507 | 0.92589 |
| | 8 | 0.94940 | 0.94842 | 0.94796 | 0.94630 | 0.94871 | **0.94954** |
| | 10 | 0.95780 | 0.95771 | 0.95459 | 0.95293 | 0.95719 | **0.95807** |
| | 12 | 0.96015 | **0.96147** | 0.95914 | 0.95719 | 0.96127 | 0.96136 |
| Image2 | 5 | 0.93647 | 0.93647 | 0.93650 | 0.93106 | 0.93647 | **0.93651** |
| | 8 | 0.95453 | 0.95499 | 0.95462 | 0.94919 | **0.95522** | 0.95467 |
| | 10 | 0.95872 | 0.96013 | 0.95853 | 0.95250 | 0.95954 | **0.96073** |
| | 12 | 0.96173 | 0.96345 | 0.96224 | 0.95771 | 0.96221 | **0.96381** |
| Image3 | 5 | 0.91768 | 0.91768 | 0.91789 | 0.91374 | 0.91771 | **0.91801** |
| | 8 | 0.94948 | 0.95066 | 0.95012 | 0.94086 | 0.95052 | **0.95093** |
| | 10 | 0.95604 | **0.95947** | 0.95561 | 0.94693 | 0.95820 | 0.95912 |
| | 12 | 0.96119 | 0.96419 | 0.96258 | 0.95046 | 0.96169 | **0.96424** |
| Image4 | 5 | **0.93398** | 0.93384 | 0.93392 | 0.93074 | 0.93384 | 0.93386 |
| | 8 | 0.95699 | **0.95705** | 0.95508 | 0.94808 | 0.95691 | 0.95668 |
| | 10 | 0.96056 | **0.96296** | 0.96129 | 0.95247 | 0.96137 | 0.96256 |
| | 12 | 0.96372 | 0.96446 | 0.96392 | 0.95410 | 0.96465 | **0.96507** |
| Image5 | 5 | **0.87714** | 0.87682 | 0.87642 | 0.87320 | 0.87682 | 0.87687 |
| | 8 | 0.91847 | 0.91933 | 0.91849 | 0.90567 | 0.91877 | **0.91938** |
| | 10 | 0.93325 | 0.93385 | 0.93102 | 0.91754 | 0.93411 | **0.93667** |
| | 12 | 0.94255 | 0.94732 | 0.94493 | 0.93006 | 0.94350 | **0.95031** |
| Image6 | 5 | 0.92420 | 0.92416 | 0.92404 | 0.92155 | 0.92425 | **0.92437** |
| | 8 | 0.95136 | 0.95161 | 0.95079 | 0.94402 | 0.952270 | **0.95255** |
| | 10 | 0.95886 | 0.959070 | 0.95556 | 0.95043 | **0.96066** | 0.96026 |
| | 12 | 0.96399 | 0.96373 | 0.96193 | 0.95793 | 0.96395 | **0.96622** |

where $I$ is the original image (with $z(i)$ is its histogram); $i = 1, 2, \ldots, L$, $L$ is a gray level); $t$ is the threshold value for extracting feature image $I$, and $u$ is calculated as follows.

$$u(a, b) = \sum_{i=a}^{b-1} iz(i) / \sum_{i=a}^{b-1} z(i) \tag{21}$$

Then the cross-entropy can be rewritten as expression follows.

$$\mathrm{D}(t) = \sum_{i=1}^{t-1} iz(i)\, log\left(\frac{i}{u(1,t)}\right) + \sum_{i=t}^{L} iz(i)\, log\left(\frac{i}{u(t, L+1)}\right) \tag{22}$$

Also, its expression can be expressed as follows.

$$\mathrm{D}(t) = \sum_{i=1}^{L} iz(i) \log(i) - \sum_{i=1}^{t-1} iz(i) \log(u(1,t)) - \sum_{i=t}^{L} iz(i) \log(u(t, L+1)) \tag{23}$$

The extension with the case of the n thresholds can apply obtained multilevel image feature extraction that can be expressed as follows.

$$\mathrm{D}(t_1, \ldots, t_n) = \sum_{i=1}^{L} iz(i)\, log(i) - \sum_{i=1}^{t_1-1} iz(i)\, log(u(1, t_1)) - \sum_{i=t_1}^{t_2-1} iz(i)\, log(u(t_1, t_2)) - \sum_{i=t_n}^{L} iz(i)\, log(u(t_n, L+1)) \tag{24}$$

The minimum cross-entropy determines the optimal threshold values, in which adding $t_0 = 1$, $t_{n+1} = L + 1$, and then the objective function can be redefined as follows.

$$\mathrm{f}(t_1, \ldots, t_n) = -\sum_{k=0}^{n} \sum_{i=t_k}^{t_{k+1}-1} iz(i) \log(u(t_k, t_{k+1})) \tag{25}$$

The optimal $n$ thresholds $\{t_1^*, \ldots, t_n^*\}$ by the MCET can be calculated as follows. $\{t_1^*, \ldots, t_n^*\} = argmin\{f(t_1, \ldots, t_n)\}$. The IMFO algorithm for image segmentation is illustrated with flowchart steps, as shown in Figure 3.

### B. EXPERIMENTAL DATA OF MULTITHRESHOLD IMAGE SEGMENTATION

Six color images are selected from the library to test the efficiency of the proposed scheme for the segmentation of

**TABLE 10.** Comparison of the obtained results of the optimization of the proposed imfo scheme with the PSO, MFO, GWO, ABC, IABC schemes for multilevel image segmentation based on a metric of the EPI.

| Image | threshold | *EPI* | | | | | |
|---|---|---|---|---|---|---|---|
| | | PSO | GWO | MFO | ABC | IABC | **IMFO** |
| Image1 | 5 | 0.92035 | 0.91882 | 0.91768 | **0.93505** | 0.91753 | 0.91951 |
| | 8 | 0.95478 | 0.95056 | 0.95144 | **0.95707** | 0.95114 | 0.95543 |
| | 10 | 0.96993 | 0.96812 | 0.96649 | 0.96276 | 0.97011 | **0.97035** |
| | 12 | 0.97581 | 0.97711 | 0.97501 | 0.96871 | 0.97522 | **0.97727** |
| Image2 | 5 | 0.93451 | 0.93451 | 0.93631 | **0.93729** | 0.93442 | 0.93702 |
| | 8 | **0.95843** | 0.95421 | 0.95585 | 0.94682 | 0.95392 | 0.95566 |
| | 10 | 0.96396 | 0.96858 | 0.96117 | 0.95660 | 0.96649 | **0.96937** |
| | 12 | 0.96678 | **0.97573** | 0.97128 | 0.96435 | 0.97321 | 0.97535 |
| Image3 | 5 | 0.86218 | 0.86218 | 0.86247 | 0.86148 | 0.86223 | **0.86251** |
| | 8 | 0.91790 | 0.91884 | 0.91769 | 0.91015 | 0.91765 | **0.92019** |
| | 10 | 0.96396 | 0.96858 | 0.96117 | 0.95660 | 0.96649 | **0.96937** |
| | 12 | 0.94778 | 0.95143 | 0.94864 | 0.92899 | 0.94839 | **0.95125** |
| Image4 | 5 | 0.97134 | 0.97253 | 0.97243 | **0.97578** | 0.97270 | 0.97219 |
| | 8 | 0.98346 | 0.98384 | 0.98558 | 0.98416 | 0.98348 | **0.98840** |
| | 10 | 0.99136 | **0.99556** | 0.99346 | 0.99036 | 0.99446 | 0.99339 |
| | 12 | 0.99764 | 0.99677 | **0.99989** | 0.99144 | 0.99824 | 0.99925 |
| Image5 | 5 | **0.85950** | 0.85865 | 0.85821 | 0.87061 | 0.85843 | 0.85932 |
| | 8 | 0.89521 | 0.89580 | 0.89497 | 0.89356 | **0.89707** | 0.89644 |
| | 10 | 0.91684 | 0.91407 | 0.91023 | 0.90790 | 0.91739 | **0.91786** |
| | 12 | 0.92987 | 0.93426 | 0.93229 | 0.92056 | 0.93486 | **0.93939** |
| Image6 | 5 | 0.95977 | 0.95966 | 0.95951 | 0.95196 | 0.95972 | **0.96019** |
| | 8 | 0.98706 | 0.98767 | 0.98765 | 0.97760 | 0.98750 | **0.98899** |
| | 10 | 0.99247 | 0.99507 | 0.99523 | 0.98735 | 0.99382 | **0.99603** |
| | 12 | 0.99947 | 1.00014 | 1.00129 | 0.99218 | 1.00154 | **1.00347** |

multilevel thresholds [45], [46]. The threshold values of the three (RGB) channels as three color components: red, green, and blue, and the threshold values are set at 5, 8, 10, and 12. Figure 4 lists the chosen picture color with separate bands (RGB) with each color image, with it being a multidimensional, multimodal model. The optimal objective function value is equal to the sum of the best objective function values of the three components. The parameter setting for the IMFO and the metaheuristic optimization algorithms are the same conditions for the experiment, e.g., the population size of all algorithms is set to 100, the maximum number of iterations is 200, the threshold number is $K = 3, 5, 8$; be set to 1 for the IMFO and MFO, $C_1 = C_2 = 1.5$ for the PSO, and the inertia weight coefficient is from 0.9 to 0.2. For GWO, the parameter setting is the same as the original algorithm $a \in [0, 2]$, $b = 1, l \in [-1, 1]$. Each algorithm runs 20 times on six images, and the average value of the segmentation index is taken.

The findings obtained from the proposed IMFO multithreshold image segmentation scheme as an interface attribute extraction are contrasted with the methods of the PSO [28], GWO [40], Artificial bee colony algorithm (ABC)[47], Improved artificial bee colony (IABC) [46], and MFO [46], schemes, respectively. Channels of the color image, then the segmented results are concatenated to form the segmented image finally.

Figure 5 displays some visually segmented images. Figure 6 demonstrates the integration of the proposed IMFO scheme with the three channels visually derived (RGB) for image 06 with thresholds set at 5. This figure also shows the analysis of IMFO scheme convergence in comparison with methods PSO, and GWO schemes. Through using different optimization algorithms in comparison means that the accuracy of the segmented image has increased with threshold changes by the proposed IMFO scheme. It is seen that the suggested IMFO scheme can deliver the coverage faster than the PSO scheme and the GWO scheme. Figure 7 depicts the image channels visually extracted (RGB) in comparison of graphs obtained under different threshold values PSO, GWO, and ABC schemes.

In order to assess the quality of the segmented image of the results obtained from the experiment, metrics of PSNR and SSIM, FSIM, and *EPI* are used to perform a comprehensive evaluation of the various algorithms in comparison. Table 6 lists the metric of evaluation parameters used to measure segmented image results.

Tables 7 to 10 list the comparison of the obtained results of the optimization of the proposed IMFO scheme with the

PSO, ABC, MFO, GWO, IABC schemes for the multilevel image segmentation based on the several aspects, e.g., PSNR, SSIM, FSIM, and EPI metrics The higher rating of the *PSNR, SSIM, FSIM, EPI* parameters, the better segmented multi-threshold results are, and the accuracy and visibility of the segmented image are as good as the original image for visible. In Tables of 7 to 10, the best values of the thresholds of the images are highlighted. From the data values from the tables, we can see that the number highlight of the six color images obtain similar segmented results belongs to the proposed scheme. Typically defines more than the other opposing algorithms, the suggested IMFO algorithm. Overall, the proposed IMFO algorithm is accurate and feasible in the resolution of the multilevel image segmentation problem.

## V. CONCLUSION

In this article, we proposed a new method of enhancing the Moth-Flame Optimization (MFO) algorithm by adjusting the flame-update positioning mode based on hybridizing the Lévy flight, adding inertia weight, and polynomial iteration feature to increase the algorithm's optimization efficiency. The improved MFO algorithm (IMFO) has been applied to the multithreshold image segmentation problem, with its minimum cross-entropy threshold segmentation method has been used to experiment. In the experimental portion, the proposed IMFO scheme outputs are evaluated using a selected series of CEC2013 benchmark functions and the multithreshold image segmentation to test the suggested scheme performance. The comparison of the results of the proposed IMFO scheme with the various algorithms in the literature under the same case situations shows that the recommended solution produces better performance than the different competing algorithms.

## REFERENCES

[1] I. Kokkinos and P. Maragos, "Synergy between object recognition and image segmentation using the expectation-maximization algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1486–1501, Aug. 2009.

[2] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, 1999, pp. 1470–1477.

[3] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.

[4] T.-T. Nguyen, T.-K. Dao, H.-Y. Kao, M.-F. Horng, and C.-S. Shieh, "Hybrid particle swarm optimization with artificial bee colony optimization for topology control scheme in wireless sensor networks," *J. Internet Technol.*, vol. 18, no. 4, pp. 743–752, 2017.

[5] T.-S. Pan, T.-K. Dao, T.-T. Nguyen, S.-C. Chu, and J.-S. Pan, "Bees and pollens with communication strategy for optimization," in *Proc. Asian Conf. Intell. Inf. Database Syst.*, vol. 9622, 2016, pp. 651–660.

[6] Y. Qiao, T.-K. Dao, J.-S. Pan, S.-C. Chu, and T.-T. Nguyen, "Diversity teams in soccer league competition algorithm for wireless sensor network deployment problem," *Symmetry*, vol. 12, no. 3, p. 445, Mar. 2020.

[7] T.-K. Dao, T.-T. Nguyen, J.-S. Pan, Y. Qiao, and Q.-A. Lai, "Identification failure data for cluster heads aggregation in WSN based on improving classification of SVM," *IEEE Access*, vol. 8, pp. 61070–61084, 2020.

[8] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, and J. C. Bansal, "A survey on parallel particle swarm optimization algorithms," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2899–2923, Apr. 2019.

[9] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.

[10] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, 1995, vol. 6, no. 4, pp. 1942–1948.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[13] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, 2009, pp. 210–214.

[14] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[15] S. C. Chu, J.-F. Chang, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *J. Inf. Sci. Eng.*, vol. 21, no. 4, pp. 809–818, 2005.

[16] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[17] S. H. H. Mehne and S. Mirjalili, "Moth-flame optimization algorithm: Theory, literature review, and application in optimal nonlinear feedback control design," in *Nature-Inspired Optimizers* (Studies in Computational Intelligence), vol. 811, S. Mirjalili, J. S. Dong, and A. Lewis, Eds. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-12127-3_9.

[18] W. K. Li, W. L. Wang, and L. Li, "Optimization of water resources utilization by multi-objective moth-flame algorithm," *Water Resour. Manage.*, vol. 32, no. 10, pp. 3303–3316, Aug. 2018.

[19] R. Barham, A. Sharieh, and A. Sleit, "Multi-moth flame optimization for solving the link prediction problem in complex networks," *Evol. Intell.*, vol. 12, no. 4, pp. 563–591, Dec. 2019.

[20] M. A. Taher, S. Kamel, F. Jurado, and M. Ebeed, "An improved moth-flame optimization algorithm for solving optimal power flow problem," *Int. Trans. Electr. Energy Syst.*, vol. 29, no. 3, p. e2743, Mar. 2019.

[21] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Comput. Vis. Image Understand.*, vol. 110, no. 2, pp. 260–280, May 2008.

[22] H. Chaur-Heh and C. Tsorng-Lin, "Analysis of evaluation metrics for image segmentation," *J. Inf. Hiding Multimedia Signal Process.*, vol. 9, no. 6, pp. 1559–1576, 2018.

[23] M. Amrehn, J. Glasbrenner, S. Steidl, and A. Maier, "Comparative evaluation of interactive segmentation approaches," in *Proc. Informatik Aktuell*, 2017, pp. 68–73.

[24] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Comput. Sci.*, vol. 65, pp. 797–806, Jan. 2015.

[25] T.-G. Ngo, T.-T. Nguyen, Q.-T. Ngo, D.-D. Nguyen, and S.-C. Chu, "Similarity shape based on skeleton graph matching," *J. Inf. Hiding Multimedia Signal Process.*, vol. 7, no. 6, pp. 1254–1264, 2016.

[26] S. Liu, X. Shen, Y. Feng, and H. Chen, "A novel histogram region merging based multithreshold segmentation algorithm for MR brain images," *Int. J. Biomed. Imag.*, vol. 2017, Jan. 2017, Art. no. 9759414.

[27] X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 131–137, Jan. 2003.

[28] P.-Y. Yin, "Multilevel minimum cross entropy threshold selection based on particle swarm optimization," *Appl. Math. Comput.*, vol. 184, no. 2, pp. 503–513, Jan. 2007.

[29] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.

[30] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[31] T.-K. Dao, J. Yu, T.-T. Nguyen, and T.-G. Ngo, "A hybrid improved MVO and FNN for identifying collected data failure in cluster heads in WSN," *IEEE Access*, vol. 8, pp. 124311–124322, 2020.

[32] A. G. Hussien, M. Amin, and M. A. El Aziz, "A comprehensive review of moth-flame optimisation: Variants, hybrids, and applications," *J. Exp. Theor. Artif. Intell.*, pp. 1–21, 2020.

[33] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, and X. Zhang, "Enhanced moth-flame optimizer with mutation strategy for global optimization," *Inf. Sci.*, vol. 492, pp. 181–203, Aug. 2019.

[34] Y. Xu, H. Chen, A. A. Heidari, J. Luo, Q. Zhang, X. Zhao, and C. Li, "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Syst. Appl.*, vol. 129, pp. 135–155, Sep. 2019.

[35] Z. Li, Y. Zhou, S. Zhang, and J. Song, "Lévy-flight moth-flame algorithm for function optimization and engineering design problems," *Math. Problems Eng.*, vol. 2016, Jan. 2016, Art. no. 1423930.

[36] L. Zhang, K. Mistry, S. C. Neoh, and C. P. Lim, "Intelligent facial emotion recognition using moth-firefly optimization," *Knowl.-Based Syst.*, vol. 111, pp. 248–267, Nov. 2016.

[37] A. A. Elsakaan, R. A. El-Sehiemy, S. S. Kaddah, and M. I. Elsaid, "An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions," *Energy*, vol. 157, pp. 1063–1078, Aug. 2018.

[38] Z. Wu, D. Shen, M. Shang, and S. Qi, "Parameter identification of single-phase inverter based on improved moth flame optimization algorithm," *Electr. Power Compon. Syst.*, vol. 47, nos. 4–5, pp. 456–469, Mar. 2019.

[39] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[40] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," *Expert Syst. Appl.*, vol. 86, pp. 64–76, Nov. 2017.

[41] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou, China Nanyang Technol. Univ. Singapore, Tech. Rep. 201212, 2013.

[42] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China, Nanyang Technol. Univ. Singapore, Tech. Rep. 635, 2013.

[43] R. Tanabe and A. Fukunaga, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1952–1959.

[44] N. R. Pal, "On minimum cross-entropy thresholding," *Pattern Recognit.*, vol. 29, no. 4, pp. 575–580, 1996.

[45] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, "Image processing with ImageJ," *Biophoton. Int.*, vol. 11, no. 7, pp. 36–42, 2004.

[46] H. Gao, Z. Fu, C.-M. Pun, H. Hu, and R. Lan, "A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm," *Comput. Electr. Eng.*, vol. 70, pp. 931–938, Aug. 2018.

[47] M.-H. Horng, "Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 13785–13791, 2011.


**TRONG-THE NGUYEN** received the Ph.D. degree in communication engineering from the National Kaohsiung University of Applied Sciences, Taiwan, in 2016. He is currently a Lecturer with the College of Information Science and Engineering, Fujian University of Technology, China, and the Department of Information Technology, Haiphong Private University, Vietnam. His current research interests include computational intelligence, signal processing, and sensor networks.

**HONG-JIANG WANG** received the B.S. degree in electrical engineering from Xihua University, China, in 2019. He is currently pursuing the M.S. degree with the College of Information Science and Engineering, Fujian University of Technology, China. His current research interests include data mining and swarm intelligence.

**THI-KIEN DAO** received the Ph.D. degree in electronic and engineering from the National Kaohsiung University of Science and Technology, Taiwan, in 2019. She is currently a Lecturer with the College of Information Science and Engineering, Fujian University of Technology. Her current research interests include computational intelligence, data mining, and sensor networks.

**JENG-SHYANG PAN** received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from The University of Edinburgh, U.K., in 1996. He is currently a Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. His current research interests include soft computing, information security, and signal processing. He joined the Editorial Board of *LNCS Transactions on Data Hiding and Multimedia Security*, the *Journal of Computers*, and the *Chinese Journal of Electronics*.

**TRUONG-GIANG NGO** received the B.S. degree in informatics, and the M.S. and Ph.D. degrees in computer science from Hanoi University, Vietnam, in 1998, 2005, and 2015, respectively. He is currently a Lecturer with the Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam. His current research interests include content-based image retrieval, intelligent image processing, multimedia systems, and computational intelligence.

**JIE YU** received the M.S. and Ph.D. degrees in automotive engineering from Fuzhou University, China, in 2009 and 2020, respectively. He is currently a Lecturer with the College of Mechanical and Automotive Engineering, Fujian University of Technology, Fuzhou, China. His current research interests include swarm intelligence, automotive engineering, and data mining.