A Collaborative Dragonfly Algorithm with Novel Communication Strategy and Application for Multi-Thresholding Color Image Segmentation

Fei-Fei Liu[1], Shu-Chuan Chu[1,2], Xiaopeng Wang[1], Jeng-Shyang Pan[1]*
[1]College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2]College of Science and Engineering, Flinders University, Australia
13210318436@163.com, scchu0803@gmail.com, 931087291@qq.com, jspan@ieee.org

Abstract

The Dragonfly Algorithm (DA) is a novel swarm intelligence algorithm with some positive applications in recent years. The algorithm simulates the basic survival ability of dragonflies to evade predators and capture prey in natural environment. The original DA algorithm converges too fast, and it is easy to fall into the local optimum, which causes the search to stagnate and the algorithm effect is not ideal. Based on above, a collaborative evolutionary dragonfly algorithm (CDA) with multi-group strategy is proposed in this paper. It uses multi-group strategy and Cauchy mutation to jointly improve the convergence speed and accuracy of the original algorithm. Image segmentation is an essential aspect of computer graphics and image processing. It has become increasingly important. This paper uses threshold technology based on the CDA algorithm to find the optimal index value under different threshold conditions. The experimental results have demonstrated that the CDA is highly competitive in terms of convergence speed and convergence accuracy DA algorithm, and CDA also performs excellent advantages in graphic segmentation experiments.

Keywords: Dragonfly algorithm, Swarm Intelligence, collaborative multi-group strategy, Image segmentation, Multi-Thresholding

1 Introduction

Since the beginning of the 20th century, swarm intelligence has drawn some researchers attention by simulating the social behavior of various creatures in nature. Some well-known algorithms play an incredibly important role in a variety of industrial, economic and social applications. Intelligent computing is part of artificial intelligence in terms of its affiliation [1-2]. Realizing the described optimization problem and solving the optimization problem is the way and method of intelligent calculation. Intelligent computing comprises three main areas: Fuzzy, Neural Network and Evolutionary Computation. The main concepts of Evolutionary Computation is to simulate Darwin's theory of biological evolution in the calculation process to solve specific ways and methods, and then to solve complex problems. Evolutionary Computation mainly includes Genetic Algorithm(GA) [3-4], Particle Swarm Optimization (PSO) [5], Cat Swarm Optimization(CSO)[6], Differential Evolution(DE) [7-8], Ant Colony Optimization (ACO) [9], Artificial Bee Colony (ABC) [10-11], Grey Wolf Optimization (GWO) [12-14], Pigeon-Inspired optimization(PIO) [15-16], Cuckoo Search(CS) [17-19], QUasi-Affine TRansformation Evolution (QUATRE) [20-22], Multiverse Optimization Algorithm (MVO) [23-24].

The purpose of optimization is to solve the optimal value of the objective function that has been constructed. Intelligent calculation is generally simple and useful, it can solve linear and non-linear problems, it also perform better in low and medium dimension. DA is a new algorithm for swarm intelligence [25]. Philip.T.Daely et al. [26] proposed a distance-based dragonfly wireless node location algorithm. Diptanu Das and others [27] used DA to solve the problem of probabilistic economic load distribution.

Image segmentation is a part of image processing [28-30]. It divides the image into several no overlap subareas. The features in the same subareas are similar, the features in different subareas are very different. A frequent method is the threshold-based segmentation method. In order to obtain better image quality after segmentation, the concept of threshold is introduced. The key is to determine the right threshold value to accurately segment the image. The optimal threshold value is determined and compared to the gray scale value of each pixel in the image. Generally, if the threshold is greater than the gray value of a certain pixel, it can be determined that the image element is the background; if the threshold is less than the gray value of a certain pixel, the image element can be determined as the object.

In order to obtain the best threshold, in this paper, the minimum cross-entropy is chosen as the objective function. The contributions of the schemes proposed in this paper are as follows:

(1) A collaborative multi-group structure is proposed to enhance the abilities to execute exploration and exploitation of the original dragonfly algorithm.

(2) The performance of the proposed algorithm was tested by selecting some benchmark functions. A comparative analysis of various algorithms is also carried out in this paper.

(3) By applying the improved algorithm and multiple comparison algorithms to multi-threshold image segmentation, the test have demonstrated that CDA has a good competitive effect.

The rest of the paper is designed as follows:

Section 2 briefly introduces the original Dragonfly algorithm and the multi-threshold image segmentation problem. Section 3 proposes a collaborative multi-group dragonfly algorithm. Section 4 states the application of the CDA in image segmentation and the image comparison analysis diagram of each algorithm. Section 5 summarizes the

* Corresponding Author: Jeng-Shyang Pan; E-mail: jspan@ieee.org
DOI: 10.53106/160792642022012301005

superiority of the proposed CDA and applies it to the field of image segmentation.

2 Related Work

The dragonfly algorithm is an intelligent optimization algorithm proposed by Seyedali Mirjalili in 2015. It simulates the basic survival abilities of dragonflies in nature to avoid natural enemies and hunt for prey.

2.1 Dragonfly Algorithm

The swarm behaviors of dragonflies includes static and dynamic behaviors. The main purpose of static behavior is to prey, dragonflies will fly in small groups in different areas and forage in small areas, achieving the effect of global exploration. The dynamic behavior is mainly for long-stance migration, dragonflies will form a large population and work together along a large direction. Once a small group of dragonflies find food, it will send a message to notify other small groups of dragonflies to come, realizing the effect of local exploitation. The dragonfly algorithm is based on the simulation of five social behaviors of dragonfly groups: separation, alignment, cohesion, attraction to food and distraction from enemy, as shown in Figure 1. The mathematical explanation of the algorithm is as follows:

(1) Separation: In order to prevent collisions with other individuals, each individual in the population will define the following equation (1).

$$Sep_i = -\sum_{j=1}^{N_j}\left(Pos - Pos_j\right) \quad \textbf{(1)}$$

where Pos is the position vector of the current dragonfly individual; Pos_j is the position vector of j-th adjacent individual; N_j is the number of individuals adjacent to the j-th dragonfly.

(2) Alignment: The velocity of the individuals flying in groups with adjacent individuals is the same, the mathematical equation (2) is as follows:

$$Ali_i = \frac{\sum_{j=1}^{N_j} Vel_j}{N_j} \quad \textbf{(2)}$$

where Ali_i is the position vector of the alignment behavior of i-th dragonfly individual; Vel_j is the velocity of j-th adjacent individual.

(3) Cohesion: Individuals tend to converge towards the center of adjacent individuals, the mathematical equation (3) is as follows:

$$Coh_i = \frac{\sum_{j=1}^{N_j} Pos_j}{N_j} - Pos \quad \textbf{(3)}$$

where Coh_i is the position vector of the cohesion behavior of the i-th dragonfly individual.

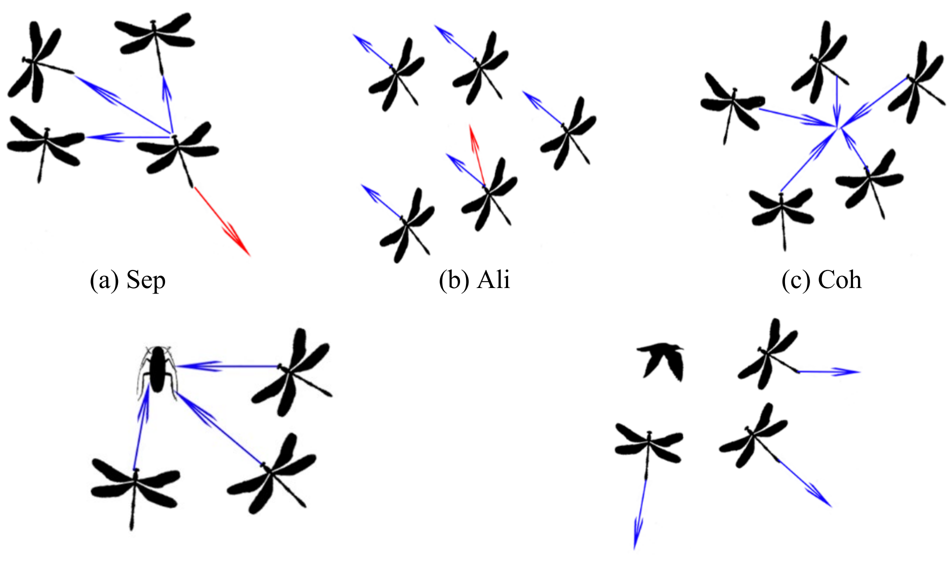
(4) Attraction to food: Individuals converge on the location of the food, the mathematical equation (4) is as follow:

$$Att_i = Pos^+ - Pos \quad \textbf{(4)}$$

where Pos^+ is the position of the food source that the best position of the iteration so far. Global optimum solution can refer to the PSO algorithm.

(5) Distraction from enemy: Each individual in order to avoid hunting by natural enemies, this behavior of avoiding predators is expressed mathematically in the equation (5) :

$$Dis_i = Pos^- - Pos \quad \textbf{(5)}$$



(a) Sep (b) Ali (c) Coh

(d) Att (e) Dis

Figure 1. Five basic behaviors of dragonfly populations

where Dis_i is the position vector that dragonfly moves to avoid natural enemies, and Pos^- is the position vector of natural enemies.

The dragonfly algorithm imitates the PSO algorithm in the process of simulating the movement of the individual, then the step vector ΔPos and position vector Pos are introduced. The step length vector represents the step length and movement direction of the dragonfly, it is similar to the velocity vector in the PSO algorithm. In this algorithm, the position of the dragonfly individual is updated and simulated, the step vector update equation (6) is as follows:

$$\begin{aligned}\Delta Pos_{t+1} = &\left(a\cdot Sep_i + b\cdot Ali_i + c\cdot Coh_i\right.\\ &\left.+d\cdot Att_i + e\cdot Dis_i\right) + \omega\cdot\Delta Pos_t\end{aligned} \quad \textbf{(6)}$$

where a represents the separation weight, b shows the alignment weight, c is the cohesion weight, d indicates the food elements, e shows the enemy elements, ω represents the inertia weight, t is the number of iterations. Sep_i, Ali_i, Coh_i, Att_i and Dis_i are the five types of correction methods mentioned above. In the process of algorithm optimization, different exploration and exploitation behaviors can be achieved through five correction methods. The area of the dragonfly is also important, so a radius r will be assumed around each dragonfly's circumference. Dragonflies will form larger populations as their radius increases.

DA is from the initial iteration of the algorithm to the pre-set number of iterations, the solutions will move to the areas of having more hopeful. When the search space diverges to the unhopeful area, the worst solution of the algorithm will be seen as a predator.

$$Pos_{t+1} = Pos_t + \Delta Pos_{t+1} \quad \textbf{(7)}$$

When the Euclidean distance between two dragonflies is less than the search radius r, the two dragonflies are considered adjacent. At this time, the dragonfly position vector update equation (7) is as above shown:

When the Euclidean distance between two dragonflies is not less than the search radius r, the two dragonflies are considered not adjacent. To improve the random behavior of dragonfly flying, the dragonfly will randomly update its position vector through the $\mathrm{Le'vy}$ flight method. The dragonfly position vector update equation (8) is as follows:

$$POS_{t+1} = POS_t + Le'vy(d)\times POS_t \quad \textbf{(8)}$$

where d represents the dimension of the position vector, t is the current iteration, Pos_t is the position vector of the dragonfly individual of the current iteration, Pos_{t+1} is the position vector of the next iteration of the dragonfly individual.

The specific information of $\mathrm{Le'vy}$ flight function is shown in the equation (9).

$$Le'vy(d) = 0.01\times\frac{r1\cdot\sigma}{|r2|^{\frac{1}{\beta}}},\quad r1\in[0,1], r2\in[0,1] \quad \textbf{(9)}$$

where β is a constant, the calculation equation (10) of σ is as follows:

$$\sigma = \left(\frac{\Gamma(1+\beta)*\sin\left(\frac{\pi*\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)*\beta*2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \quad \textbf{(10)}$$

where the equation (11) for Γ function is as follows:

$$\Gamma(x) = (x-1)! \quad \textbf{(11)}$$

2.2 Multi-Threshold Segmentation

Multi-threshold segmentation can segment multiple regions in an image by setting multiple thresholds. It is widely used due to its simple implementation and low computational effort [31-34]. Multi-threshold image segmentation by determining n threshold values to form a threshold vector $T = \{t_1, t_2, \dots t_n\}$. The defined M threshold divides the image into $n+1$ sub-regions. The scope of subregion 1 is $\{0,\dots,t_1\}$, the scope of subregion 2 is $\{t_1,\dots,t_2\}$, ..., the scope of subregion $M+1$ is $\{t_n,\dots,F\}$. Selection of a global optimum threshold vector $T = \{t_1^*, t_2^*, \dots t_n^*\}$ by using the minimum cross-entropy segmentation method. Therefore, the optimal threshold vector T^* is given by the following equation (12):

$$\{t_1^*, t_2^*, \dots t_n^*\} = argmin\{f(t_1, t_2, \dots t_n)\} \quad \textbf{(12)}$$

A few supplements to the above equation, Subject to $0 < t_1 < t_2 < \dots < t_n < F$,where f is the multi-threshold objective function for image segmentation. Find the targe vector with the minimum fitness value T to be the optimal threshold vector obtained from the image. The optimal threshold for image segmentation can be determined using minimum cross-entropy. Cross entropy is used to measure the information theoretical distance between two distributions P and Q , let $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_n\}$ be two similar probability distributions. The cross entropy is calculated as follows equation (13):

$$D(P,Q) = \sum_{i=1}^{n} p_i \cdot \log\frac{p_i}{q_i} \quad \textbf{(13)}$$

The image is segmented using the minimum cross-entropy method(MCET) and the original image is marked as I. Compare each pixel grey value in the original image with the optimal threshold separately, the two segmented image equation (14) are obtained as follows:

$$I_s(x,y) = \begin{cases} u(1,t), & I(x,y) < t \\ u(t,F+1), & I(x,y) \geq t \end{cases} \quad \textbf{(14)}$$

where I is the source image, t is the threshold set by the image. $z(i)$ is the image pixel histogram $i=(1,2,\ \ldots\ ,F,\ F\ is\ the\ number\ of\ pixel\ gray\ levels)$, where μ is calculated as equation (15):

$$u(a,b)=\frac{\sum_{i=1}^{b-1} iz(i)}{\sum_{i=1}^{b-1} z(i)} \quad \textbf{(15)}$$

The cross-entropy required to segment the image by a threshold value can be calculated as follows equation (16):

$$D(t)=\sum_{i=1}^{t-1} iz(i)\cdot\log\left(\frac{i}{u(1,t)}\right)+\sum_{i=t}^{F} iz(i)\cdot \log\left(\frac{i}{u(t,F+1)}\right) \quad \textbf{(16)}$$

It can also be expressed in another form as follows equation (17):

$$D(t)=\sum_{i=1}^{F} iz(i)\cdot\log(i)-\sum_{i=1}^{t-1} iz(i)\cdot\log(u(1,t)) -\sum_{i=\mathrm{t}}^{F} iz(i)\cdot\log(u(t,F+1)) \quad \textbf{(17)}$$

When n threshold values are required to achieve multiple image segmentation, the equation (18) takes the following form.

$$D(t_1,\ldots,t_n)=\sum_{i=1}^{F} iz(i)\cdot\log(i)-\sum_{i=1}^{t_1-1} iz(i)\cdot\log\left(u\left(1,t_1\right)\right) -\sum_{i=t_1}^{t_2-1} iz(i)\cdot\log\left(u\left(t_1,t_2\right)\right)-\cdots -\sum_{i=t_n}^{F} iz(i)\cdot\log\left(u\left(t_n,F+1\right)\right) \quad \textbf{(18)}$$

The optimal threshold can be determined by the minimum cross-entropy, which can be added $t_0=1, t_{n+1}=F+1$. The minimum cross entropy objective function is defined as equation (19):

$$f\left(t_1,\ldots,t_n\right)=-\sum_{k=0}^{n}\sum_{i=t_k}^{t_{k+1}-1} iz(i)\cdot\log\left(u\left(t_k,t_{k+1}\right)\right) \quad \textbf{(19)}$$

3 Collaborative Dragonfly Algorithm (CDA)

The original DA algorithm converges too fast and is easy to fall into the local optimum, which leads to the stagnation of the search, the effect of the algorithm is not satisfied. In this section, a new communication strategy based on the original DA algorithm will be described. The collaborative multi-group framework has more hopeful convergence speed and higher accuracy than the original algorithm. In the collaborative multi-group strategy, the entire dragonfly population is divided into G groups. In the iterative process, each subpopulation does not affect each other, and it remains relatively independent in the evolutionary calculation process. Different subpopulations will only carry out inter-species communication only when they meet the conditions for communication, which can strengthen the advantages of cooperation between groups. Each subpopulation will find an optimal food position, which plays an important role in the global optimal food position obtained by the entire population. The worst dragonfly in the subpopulation will be replaced by the best dragonfly in the different groups. In the end, the relatively worst solution of the entire population will be deleted. The fast search ability of the algorithm is guaranteed. Then a better solution can be obtained. The way of communication is shown in Figure 2.

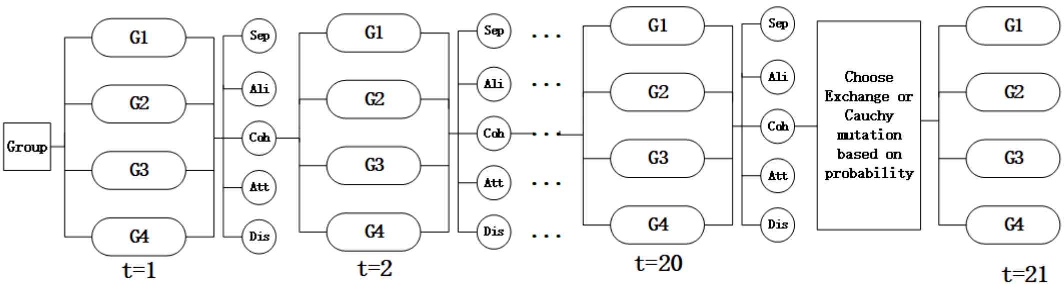


Figure 2. collaborative multi-group structure communication strategy

3.1 Communication strategy

This article introduces a collaborative multi-group framework, and introduces its importance and feasibility. In the original DA algorithm, it can also be seen as a simple multi-group framework. In principle, DA can be regarded as N_p particles divided into G groups. Communication takes place with every generation, The communication strategy is that $N-1$ particles with poor particles are approaching and learning from the best particles. In addition, it can also be seen as N_p particles divided into one group, only communicate with in the group, the communication strategy is the same as above. The collaborative multi-group framework introduced in this paper combines the above two

DA multi-group methods and is essentially similar in principle.

In this paper, CDA uses two communication strategies. No matter which strategy is used, in order to improve the randomness of the algorithm, this paper will generate a random number. Judgement is made using R as the breakpoint: if the random number is less than R then the first strategy is used, otherwise the second strategy is used. The first is: N_p dragonfly individuals are divided into groups G and each subgroup g is composed of N_p / G dragonfly individuals. N_p / G is always less than N_p, and the number of subgroups is always not greater than G. During the iteration process, replace the position of the worst dragonfly individual in the g group with the position of the best dragonfly individual in the $g+1$ group. Using the above mechanism in turn, the position of the worst dragonfly individual in group G is replaced with the position of the best dragonfly individual in the first group. The exchange strategy between groups is shown in Figure 2. The second is a strategy proposed by Cauchy's mutation, the algorithm flow of CDA is shown in pseudo 1.

Algorithm 1 Pseudo-codes of the CDA algorithm

Randomly initialize the position of the N_p dragonfies Pos_i $(i = 1, 2, ..., Np)$
Initialize step vectors ΔPos_i $(i = 1, 2, ..., Np)$
Initialize communication cycle CC
Initialize number of groups G
Initialize the communication threshold R
while $t \leq T$ **do**
    for i=1 to G **do**
        Calculate the objective values of each group dragonfies
        Update the position of food source and enemy
        Update $a,\ b,\ c,\ d,\ e,\ and\ \omega$
        Calculate Sep, Ali, Coh, Att, and Dis using Equation (1) to (5)
        Update neighbouring radius
        if a dragonfly has at least one neighbouring dragonfly **then**
            Update velocity vector using Equation (6)
            Update position vector using Equation (7)
        else
            Update position vector using Equation (8)
        end if
        Check and correct the new positions based on the boundaries of variables
    end for
    if $mod(t, CC) = 0$ **then**
        if $rand() \leq R$ **then**
            Select each worse and best dragonfies in $i - th$ group
            Replace the worst in each group with the best in the next group
            Replace the worst in the last group with the best in the first group
        else
            The second communication strategy of Cauchy mutation
        end if
    end if
end while

3.2 Cauchy mutation

Cauchy mutation can be used to randomly change the position of individual dragonflies, improving the searchability of the algorithm. It is widely used in other algorithms, such as Fish Migration Optimization (FMO) [35-36]. In the algorithm, the Cauchy mutation operator is applied to the state of the artificial fish to enhance the global search ability of the fish swarm. In order to obtain the global optimal value, Cauchy mutation is applied on the obtained optimal particles. The specific description equation (20) of Cauchy mutation is as follows:

$$\text{SalpPositions}_{mute} = \text{SalpPositions} \times (1 + Cauchy(0,1)) \tag{20}$$

The one-dimensional Cauchy density function is mainly concentrated near the origin, and the one-dimensional density function equation (21) is determined as:

$$f(x, \delta, \mu) = \frac{1}{\pi} \cdot \frac{\delta}{\delta^2 + (x - \mu)^2}, \quad -\infty < x < \infty \tag{21}$$

where $\delta = 1$, $\mu = 0$, the standard one-dimensional Cauchy density function equation (22) at the center of the origin is expressed as follows:

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{x^2 + 1}, \quad -\infty < x < \infty \tag{22}$$

In equation (23), $Cauchy(0,1)$ is explained as follows:

$$Cauchy(0,1) = \tan[(\xi - 0.5) \cdot \pi], \quad \xi \in U[0,1] \tag{23}$$

Through the description and summary of the above Cauchy mutation, the second strategy proposed in this paper uses the Cauchy mutation method. In CDA, the dragonfly populations are grouped equally in number, and then in any group, the position vector of the j-th dimension of the i-th dragonfly use the Cauchy mutation, and then the position vector of the dragonfly is randomly updated to increase the diversity of the population. Therefore, the second strategy equation (24) proposed in this paper is as follows:

$$\begin{aligned} group(g) \cdot Pos(d,i) &= group(g) \cdot Pos(d,i) \\ &+ group(g) \cdot Pos(d,i) * \tan((rand - 0.5) * \pi) \end{aligned} \tag{24}$$

where d represents the dimension of the space, i is the i-th dragonfly individual in the g-th group and $rand \in [0,1]$.

4 Experimental analysis

In this part, the collaborative evolutionary dragonfly algorithm is examined on the test function in this paper, and tested on the multi-threshold color image segmentation. It evaluates the performance of the multi-group CDA to prove the effectiveness of the improvement.

4.1 Experimental results of a collaborative dragonfly algorithm

In this section, this paper uses 23 test functions to test to confirm the performance of the proposed multi-group algorithm CDA. Where the basic parameters of these algorithms are shown in Table 4.

Table 1 to Table 3 describes the relevant information of the 23 test functions [5]. F1 to F7 are unimodal functions. For this functions, there is only one global extremum, which can be used to test the local search ability of the algorithm. From F8 to F13 are multimodal function, with a global extremum and multiple local optimal values. In this type of functions, it can be used to test the local search ability of the algorithm. From F14 to F23 are other dimensional functions, there are only a few local minimums and features with fewer dimensions. Therefore, this paper proposes that the new communication strategy performs better in the case of low latitudes and many local optimal values.

Comprehensively test the performance of the CDA algorithm, and record the optimal value, mean and standard deviation obtained by 23 test functions. By defining “win”, “draw”, and “lose” three characters to represent the best, draw, or lose selected by comparing CDA with other algorithms. The performance of the algorithm is mainly analyzed from the perspective of the mean. The smaller the mean, the better the accuracy and performance of the reflected algorithm.

From the data in Table 6 and Table 7, the proposed CDA algorithm is better than other algorithms. From the point of view of the mean value, in Table 6, compared with the original DA algorithm, the proposed CDA algorithm obtained 19 better performances in 23 test functions, 4 near performances, and 0 worse performances. The proposed CDA algorithm obtained 15 better performances, 2 near performances, and 6 worse performances in 23 test functions compared to the SCA algorithm.

In Table 7, compared with MVO algorithm, the CDA algorithm obtained 18 better performances, 4 near performances, and 1 worse performance in 23 test functions. Compare CDA with SCA and MVO algorithms. Since DA and these two algorithms are both proposed by Seyedali Mirjalili, compared with these two algorithms, the experimental results are more convincing. The proposed CDA algorithm obtained 17 better performances in 23 test functions, 1 near performance, and 5 worse performances compared to the PSO algorithm. Compared with PPSO algorithm, the CDA algorithm obtained 18 better performances, 4 near performances, and 1 worse performance in 23 test functions.

Figure 3 shows the convergence curves of the optimal values of these six algorithms. It can be seen that the performance of the proposed CDA in the test function f5, f6, f8, f11, f13, f14, f15, f20 is better than other algorithms.

All in all, the CDA algorithm is more competitive than other algorithms.

4.2 multi-threshold color image segmentation problem based CDA

In this section, Figure 4 shows eight original color images to test the performance of the algorithm proposed in this paper. This paper chooses the minimum cross-entropy threshold as the objective function to obtain the multi-threshold of the color image. Each image includes three representative brands(RGB). To be fair, each algorithm runs 30 times for each color image, the maximum number of iterations is set to 100, the initial solution number is set to 50.

In this paper, in order to clearly show the graphics segmentation effect of the proposed CDA algorithm, the segmentation information of Img1 is shown in Figure 5. The threshold is selective to prove that CDA algorithm is excellent in this paper. In the process of selecting the threshold, because there are too many choices of the threshold. The single threshold 1 was removed, and the three thresholds of 4, 7 , 10 were selected in increments of 3.

The information thresholds of the three channels (RGB) are set to 4, 7 and 10. Then the MCET function is optimized according to the pixel histogram information of each band, and the optimized multi-threshold value is obtained.

In this paper, the information thresholds of the three channels(RGB) are set to 4, 7 and 10, apply to the CDA and DA, SCA, PSO, PPSO and MVO algorithms respectively. By segmenting each channel, and then connecting the segmented images to form the final segmented image. Figure 6 to Figure 9 shows the image after segmentation (Img.2- 8), it is found that the larger the number of thresholds, the segmentation quality of the image is better.

Table 1. Unimodal test function

No	Function expression	Search space	Dimension	TM
1	$f_1(y)=\sum_{No}^{j=1} y_j^2$	[-100, 100]	30	0
2	$f_2(y)=\sum_{No}^{j=1}\lvert y_j\rvert+\prod_{No}^{j=1}\lvert y_j\rvert$	[-10, 10]	30	0
3	$f_3(y)=\sum_{No}^{j=1}\left(\sum_{j}^{k=1} y_k\right)^2$	[-100, 100]	30	0
4	$f_4(y)=max_j\lvert y_j\rvert, j\in[1,m]$	[-100, 100]	30	0
5	$f_5(y)=\sum_{No-1}^{j=1}\left[100\left(y_{j+1}-y_j^2\right)^2+\left(y_j-1\right)^2\right]$	[-30, 30]	30	0
6	$f_6(y)=\sum_{No}^{j=1}\left([y_j+0.5]\right)^2$	[-100, 100]	30	0
7	$f_7(y)=\sum_{No}^{j=1} j*y_j^2+\mathrm{rand}[0,1)$	[-1.28, 1.28]	30	0

Table 2. Multimodal test function

No	Function expression	Search space	Dimenon	TM
8	$f_8(y)=\sum_{No}^{j=1} -y_j*\sin\left(\sqrt{\lvert y_j\rvert}\right)$	[-500, 500]	30	-12569
9	$f_9(y)=\sum_{No}^{j=1}\left[y_j^2-10*cos\left(2\pi y_j\right)+10\right]$	[-5.12,5.12]	30	0
10	$f_{10}(y)=-20*\exp\left(-0.2\sqrt{\frac{1}{N_O}\sum_{No}^{j=1}y_j^2}\right)$ $-\exp\left(\frac{1}{N_O}\sum_{No}^{j=1}\cos\left(2\pi y_j\right)+20+2.718\right)$	[-32, 32]	30	0
11	$f_{11}(y)=\frac{1}{4000}*\sum_{No}^{j=1}y_j^2-\Pi_{j=1}^{No}\cos\left(\frac{y_j}{\sqrt{j}}\right)+1$	[-600,600]	30	0
12	$f_{12}(y)=\frac{\pi}{No}*\{10*\sin\left(\pi y_1\right)+$ $\sum_{No-1}^{j=1}\left(y_j-1\right)^2\left[1+10*\sin^2\left(\pi y_{j+1}\right)\right]+\left(y_{No}-\right.$ $+\sum_{No}^{j=1}u\left(y_j,10,100,4\right)$ $y_j=1+\frac{y_j+1}{4}*u\left(z_l,a,k,m\right)=\begin{cases}k\left(y_j-a\right),\\ 0,-a<y_j\\ k\left(-y_j-a\right).\end{cases}$	[-50, 50]	30	0

13	$f_{13}(y)=0.1*$ $\left\{\sin^2(3\pi y_1)+\sum_{No}^{j=1}(y_j-1)^2\left[1+\sin^2(3\pi y_j+1\right.\right.$ $\left.\left.+(y_{No}-1)^2\left[1+\sin^2(2\pi y_{No})\right]\right\}$ $+\sum_{No}^{j=1}u(y_j,10,100,4)$	[-50, 50]	30	0

Table 3. Fixed dimension function

No	Function expression	Search space	Dimension	TM
14	$f_{14}(y)=\left(\frac{1}{500}*\sum_{25}^{j=1}\frac{1}{j+\sum_{2}^{k=1}(z_k-a_{kj})^6}\right)^{-1}$	[-65,65]	2	1
15	$f_{15}(y)=\sum_{11}^{j=1}\left[a_j-\frac{y_1(b_j^2+b_jy^2)}{b_j^2+b_jy^3+y^4}\right]^2$	[-5, 5]	4	0.00030
16	$f_{16}(y)=4y_j^2-2.1y_j^4+\frac{1}{3}y_j^6$ $+y_jy_2-4y_2^2+4y_2^4$	[-5, 5]	2	-1.0316
17	$f_{17}(y)=\left(y_2-\frac{5.1}{4\pi^2}y_j^2+\frac{5}{\pi}y_j-6\right)^2$ $+10\left(1-\frac{1}{8\pi}\right)\cos y_j+10$	[-2, 2]	2	0.398
18	$f_{18}(y)=\left[1+(y_1+y_2+j)^2*\right.$ $\left.(19-14y_1+3y_1^2-14y_2+6y_1y_2+3y_2^2)\right]$ $*\left[30+(2y_1-3y_2)^2*\right.$ $\left.(18-32y_1+12y_1^2+48y_2-36y_1y_2+27y_2^2)\right.$	[1, 3]	2	3
19	$f_{19}(y)=-\sum_{4}^{j=1}c_j*exp\left(-\sum_{3}^{k=1}a_{jk}(y_k-p_{jk})^2\right)$	[0, 1]	3	-3.86
20	$f_{20}(y)=-\sum_{4}^{j=1}c_j*exp\left(-\sum_{6}^{k=1}a_{jk}(y_k-p_{jk})^2\right)$	[0, 10]	6	-3.32
21	$f_{21}(y)=-\sum_{5}^{j=1}\left[(y-a_j)(y-a_j)^T+c_j\right]^{-1}$	[0, 10]	4	-10.1532
22	$f_{22}(y)=-\sum_{7}^{j=1}\left[(y-a_j)(y-a_j)^T+c_j\right]^{-1}$	[0, 10]	4	-10.4028
23	$f_{23}(y)=-\sum_{10}^{j=1}\left[(y-a_j)(y-a_j)^T+c_j\right]^{-1}$	[0, 10]	4	-10.5363

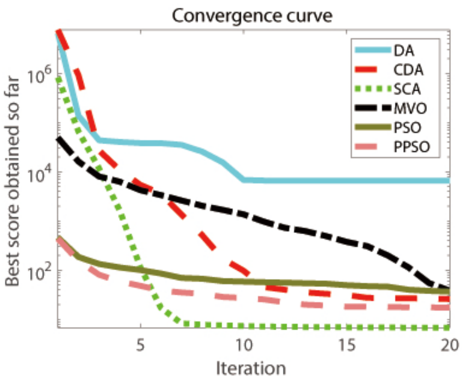
Table 4. Parameter Setting

Algorithm	Parameter
CDA	$G=4, R=20,40,\ldots,1000, Pop=100, Iteration=1000$
DA	$Pop=100, Iteration=1000$
SCA	$a=2, r_1=2$ to $0, Pop=100, Iteration=1000$
PSO	$c=2.0, \omega=0.9, V_{\min}=-10, V_{\max}=10, Pop=100, Iteration=1000$
PPSO	$c=2.0, \omega=0.9, G=4, R=20,40,\ldots,1000, V_{\min}=-10, V_{\max}=10, Pop=100$
MVO	$\omega=6, W_{\min}=0.2, W_{\max}=1, Pop=100, Iteration=1000$

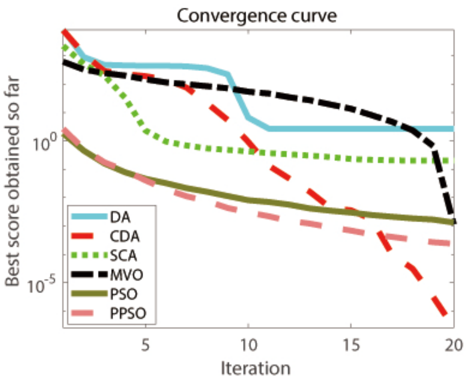
Algorithm belongs to an overall trend, and there is randomness at the same time. Therefore, there are certain differences in the performance of each image, which will cause some changes in the results of the experiment. For example, the threshold is not very suitable for the experimental results generated by the optimization algorithm, which will directly lead to the use of the optimization algorithm to not get good experimental results. This paper uses PSNR [37], SSIM [38], FSIM [39], SS-SSIM to test the difference between the image obtained after the optimization algorithm and the original image. The detailed information and equation of the four indicators are shown in Table 5.

For these four parameters, the larger the value obtained, the higher the image quality after segmentation. Table 8 to Table 11 shows all the experimental data. From the data of PSNR, SSIM, FSIM, MS-SSIM, CDA has 12, 14, 15, 15 better performances than other algorithms. The CDA algorithm has achieved an advantage of more than half on four indicators of eight images with 3 thresholds. It can be seen that the CDA algorithm with a collaborative multi-group framework performs satisfied compared to other algorithms.

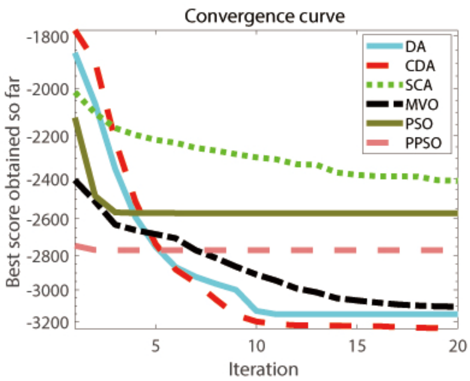
For these three thresholds (4, 7, 10), compared with other algorithms, the CDA algorithm proposed in this paper can usually get better image segmentation quality. In summary, the CDA algorithm with a collaborative multi-group framework can solve the color image segmentation problem more effectively and feasible.



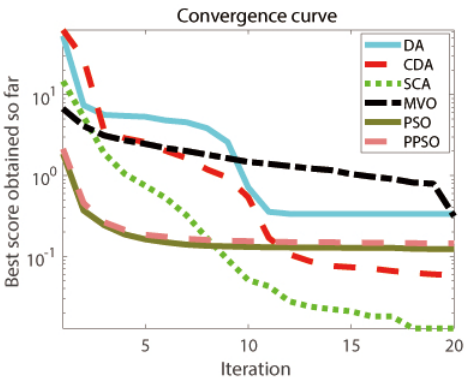
(a) Function 5



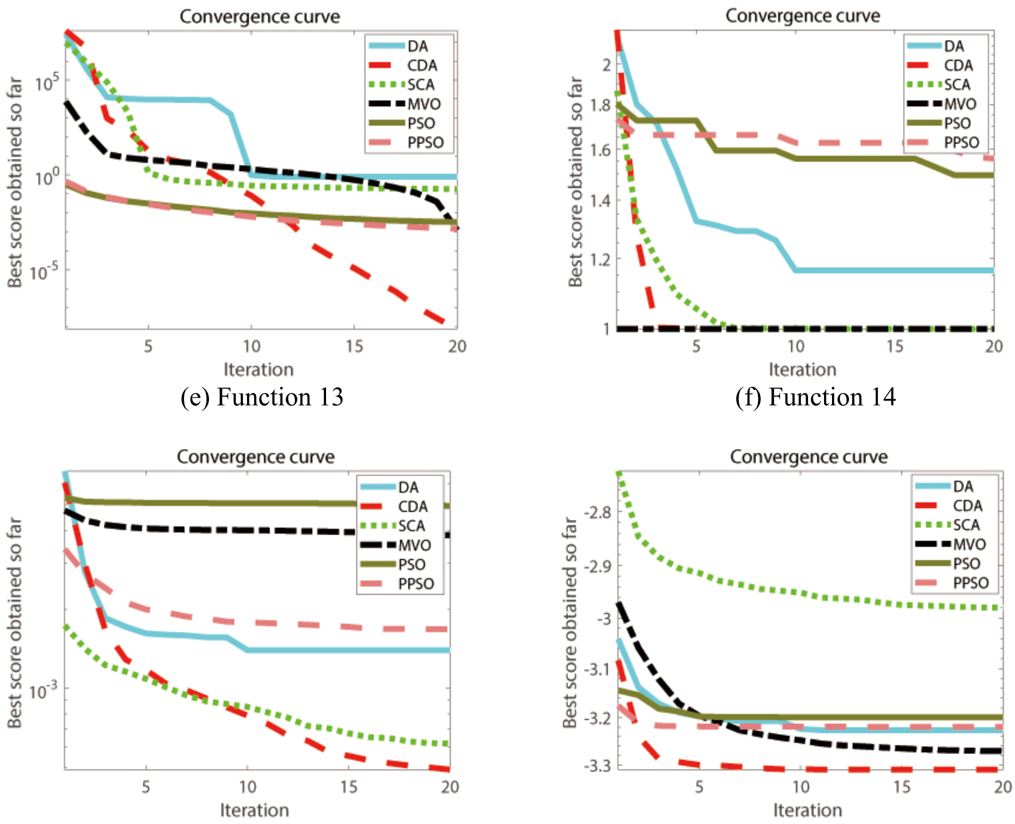
(b) Function 6



(c) Function 8



(d) Function 11



(e) Function 13

(f) Function 14

(g) Function 15

(h) Function 20

Figure 3. Performance for CDA, DA, SCA, PSO, PPSO and MVO under test functions

Table 5. Metrics used to measure evaluation parameters of segmented image results

	Parameters	**Equation**	**Remarks**
1.	PSNR	$PSNR = 20\log_{10}\dfrac{255}{\sqrt{MSE}}$	PSNR represents that the proportional relationship is between the maximum pixel value and MSE.
2.	SSIM	$SSIM(I,I) = \dfrac{(2\mu_1\mu_1\cdot+C_1)(2\sigma_{11}\cdot+C_2)}{(\mu_1^2+\mu_1^2+C_1)(\sigma_1^2+\sigma_1^2+C_2)}$	SSIM is a similarity parameter of two different image structures.
3.	FSIM	$FSIM = \dfrac{\sum_{i=1}^{N} S_L(D)PC_{\max}(D)}{\sum_{i=1}^{N} PC_{\max}(1)}$	FSIM is a similarity parameter of two different image features.
4.	MS-SSIM	$MS-SSIM(Z,K) = [l_M(Z,K)]^{\alpha_M}\prod_{i=1}^{M}[s_i(Z,K)]^{\beta_i}[z_i(Z,K)]^{\gamma_i}$	MS-SSIM is the average structural similarity, which calculates the structural similarity SSIM of the corresponding blocks of the image, and uses the average value as the structural similarity measure of the two images.

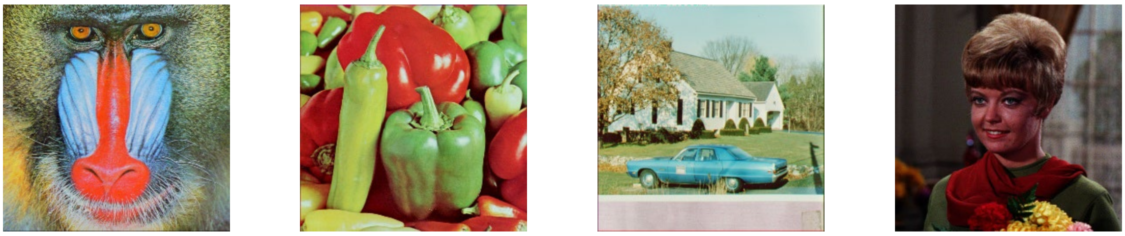
Table 6. Test comparative results of the DA, SCA and proposed CDA algorithm for selected 23 functions

Func Name	DA			SCA			CDA		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
F1	0	2.38×10^{0}	5.82×10^{0}	3.67×10^{2}	8.48×10^{3}	6.45×10^{3}	3.14×10^{-24}	2.59×10^{-13}	1.13×10^{-12}
F2	0	3.72×10^{0}	4.35×10^{0}	2.12×10^{-4}	4.28×10^{0}	4.62×10^{0}	0	9.76×10^{-11}	3.86×10^{-10}
F3	0	3.78×10^{2}	1.27×10^{3}	8.03×10^{-25}	3.78×10^{-15}	1.75×10^{-14}	1.17×10^{-11}	1.63×10^{-6}	6.47×10^{-6}
F4	0	1.11×10^{0}	1.71×10^{0}	2.58×10^{-16}	2.63×10^{-12}	7.87×10^{-12}	4.23×10^{-8}	2.49×10^{-3}	1.14×10^{-2}
F5	8.52×10^{0}	7.08×10^{3}	2.28×10^{4}	5.78×10^{0}	6.64×10^{0}	3.42×10^{-1}	3.15×10^{0}	7.79×10^{0}	8.70×10^{0}
F6	1.21×10^{-15}	2.01×10^{0}	4.18×10^{0}	7.62×10^{-2}	2.32×10^{-1}	1.16×10^{-1}	6.67×10^{-11}	5.56×10^{-8}	1.17×10^{-7}
F7	1.45×10^{-4}	6.11×10^{-3}	5.44×10^{-3}	2.85×10^{-5}	3.88×10^{-4}	3.40×10^{-4}	2.78×10^{-4}	3.07×10^{-3}	2.57×10^{-3}
F8	-3.89×10^{3}	-3.15×10^{3}	3.35×10^{2}	-2.73×10^{3}	-2.41×10^{3}	1.77×10^{2}	-3.83×10^{3}	-3.24×10^{3}	2.78×10^{2}
F9	0	1.22×10^{1}	1.17×10^{1}	0	2.10×10^{-10}	1.15×10^{-9}	0	4.13×10^{0}	3.67×10^{0}
F10	8.88×10^{-16}	1.53×10^{0}	1.50×10^{0}	8.88×10^{-16}	4.32×10^{-15}	6.49×10^{-16}	4.44×10^{-15}	5.09×10^{-8}	1.27×10^{-7}
F11	0	3.35×10^{-1}	3.47×10^{-1}	0	1.28×10^{-2}	5.48×10^{-2}	0	5.81×10^{-2}	3.62×10^{-2}
F12	2.04×10^{-6}	5.58×10^{-1}	7.45×10^{-1}	8.79×10^{-3}	4.38×10^{-2}	2.01×10^{-2}	5.10×10^{-10}	6.23×10^{-4}	3.40×10^{-3}
F13	1.66×10^{-3}	8.02×10^{-1}	1.51×10^{0}	8.33×10^{-2}	1.84×10^{-1}	5.96×10^{-2}	8.71×10^{-11}	7.75×10^{-9}	1.12×10^{-8}
F14	9.98×10^{-1}	1.16×10^{0}	3.77×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	3.16×10^{-6}	9.98×10^{-1}	9.98×10^{-1}	2.06×10^{-16}
F15	4.88×10^{-4}	1.40×10^{-3}	5.19×10^{-4}	3.12×10^{-4}	6.17×10^{-4}	3.43×10^{-4}	3.07×10^{-4}	4.91×10^{-4}	3.50×10^{-4}
F16	-1.03×10^{0}	-1.03×10^{0}	2.28×10^{-12}	-1.03×10^{0}	-1.03×10^{0}	8.85×10^{-6}	-1.03×10^{0}	-1.03×10^{0}	5.26×10^{-16}
F17	3.98×10^{-1}	3.98×10^{-1}	0	3.98×10^{-1}	3.98×10^{-1}	5.09×10^{-4}	3.98×10^{-1}	3.98×10^{-1}	0
F18	3.00×10^{0}	3.00×10^{0}	1.04×10^{-5}	3.00×10^{0}	3.00×10^{0}	1.09×10^{-6}	3.00×10^{0}	3.00×10^{0}	5.86×10^{-13}
F19	-3.86×10^{0}	-3.86×10^{0}	1.73×10^{-3}	-3.86×10^{0}	-3.86×10^{0}	3.28×10^{-3}	-3.86×10^{0}	-3.86×10^{0}	5.14×10^{-15}
F20	-3.32×10^{0}	-3.23×10^{0}	8.65×10^{-2}	-3.24×10^{0}	-2.98×10^{0}	2.25×10^{-1}	-3.32×10^{0}	-3.31×10^{0}	3.62×10^{-2}
F21	-1.02×10^{1}	-9.88×10^{0}	9.57×10^{-1}	-8.53×10^{0}	-4.31×10^{0}	1.87×10^{0}	-1.02×10^{1}	-1.02×10^{1}	1.68×10^{-10}
F22	-1.04×10^{1}	-1.02×10^{1}	9.71×10^{-1}	-8.13×10^{0}	-5.62×10^{0}	1.08×10^{0}	-1.04×10^{1}	-1.04×10^{1}	2.39×10^{-4}
F23	-1.05×10^{1}	-1.04×10^{1}	4.60×10^{-1}	-8.18×10^{0}	-5.54×10^{0}	9.99×10^{-1}	-1.05×10^{1}	-1.05×10^{1}	1.93×10^{-5}
Win	5	19	22	17	15	15	–	–	–
Draw	11	4	1	3	2	0	–	–	–

Table 7. Test comparative results of the MVO, PSO, PPSO and proposed CDA algorithm for selected 23 functions

Func Name	MVO			PSO			PPSO		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
F1	3.78×10^{-4}	1.17×10^{-3}	4.01×10^{-4}	3.04×10^{-4}	1.54×10^{-3}	1.30×10^{-3}	2.07×10^{-5}	1.93×10^{-4}	1.58×10^{-4}
F2	4.51×10^{-3}	9.85×10^{-3}	3.23×10^{-3}	0	0	0	0	0	0
F3	1.05×10^{-3}	3.52×10^{-3}	2.15×10^{-3}	1.43×10^{0}	2.52×10^{-2}	1.56×10^{-2}	1.41×10^{-4}	5.53×10^{-3}	5.62×10^{-3}
F4	1.34×10^{-2}	2.26×10^{-2}	5.59×10^{-3}	2.14×10^{-2}	4.71×10^{-2}	2.43×10^{-2}	4.69×10^{-3}	2.11×10^{-2}	1.06×10^{-2}
F5	3.45×10^{0}	4.48×10^{1}	8.75×10^{1}	2.75×10^{0}	3.87×10^{1}	8.34×10^{1}	3.58×10^{0}	2.56×10^{1}	4.11×10^{1}
F6	3.44×10^{-4}	1.18×10^{-3}	4.77×10^{-4}	2.76×10^{-4}	1.36×10^{-3}	9.58×10^{-4}	5.99×10^{-5}	2.38×10^{-4}	1.58×10^{-4}
F7	1.04×10^{-4}	6.02×10^{-4}	3.45×10^{-4}	9.44×10^{-4}	5.04×10^{-3}	3.93×10^{-3}	8.08×10^{-5}	1.71×10^{-3}	1.12×10^{-3}
F8	-3.85×10^{3}	-3.10×10^{3}	3.03×10^{2}	-3.26×10^{3}	-2.57×10^{3}	2.63×10^{2}	-3.32×10^{3}	-2.77×10^{3}	2.44×10^{2}
F9	5.97×10^{0}	1.22×10^{1}	5.06×10^{0}	1.01×10^{-1}	7.51×10^{0}	6.53×10^{-0}	3.35×10^{-4}	2.87×10^{0}	6.60×10^{0}
F10	9.36×10^{-3}	1.48×10^{-2}	2.83×10^{-3}	3.00×10^{-2}	1.66×10^{-1}	3.72×10^{-1}	3.90×10^{-3}	2.27×10^{-2}	1.33×10^{-2}
F11	1.05×10^{-1}	3.15×10^{-1}	1.71×10^{-1}	3.52×10^{-2}	1.23×10^{-1}	$7. 61 \times 10^{-2}$	2.23×10^{-2}	1.45×10^{-1}	8.61×10^{-2}
F12	9.07×10^{-6}	1.04×10^{-2}	5.68×10^{-2}	9.02×10^{-5}	5.85×10^{-4}	4.65×10^{-4}	9.20×10^{-6}	9.56×10^{-5}	5.66×10^{-5}
F13	4.24×10^{-5}	1.23×10^{-3}	3.40×10^{-3}	9.15×10^{-5}	3.33×10^{-3}	5.11×10^{-3}	1.24×10^{-4}	1.45×10^{-3}	3.01×10^{-3}
F14	9.98×10^{-1}	9.98×10^{-1}	2.56×10^{-12}	9.98×10^{-1}	1.49×10^{0}	6.77×10^{-1}	9.98×10^{-1}	1.56×10^{0}	7.22×10^{-1}
F15	3.08×10^{-4}	3.83×10^{-3}	7.52×10^{-3}	7.88×10^{-4}	4.95×10^{-3}	7.33×10^{-3}	5.95×10^{-4}	1.68×10^{-3}	3.54×10^{-3}
F16	-1.03×10^{0}	-1.03×10^{0}	4.24×10^{-8}	-1.03×10^{0}	-1.03×10^{0}	6.16×10^{-7}	-1.03×10^{0}	-1.03×10^{0}	1.30×10^{-7}
F17	3.98×10^{-1}	3.98×10^{-1}	6.97×10^{-8}	3.98×10^{-1}	3.98×10^{-1}	5.54×10^{-4}	3.98×10^{-1}	3.98×10^{-1}	6.01×10^{-4}
F18	3.00×10^{0}	3.00×10^{0}	1.98×10^{-7}	3.00×10^{0}	3.00×10^{0}	3.93×10^{-6}	3.00×10^{0}	3.00×10^{0}	2.50×10^{-6}
F19	-3.86×10^{0}	-3.86×10^{0}	5.47×10^{-8}	-3.86×10^{0}	-3.86×10^{0}	2.72×10^{-3}	-3.86×10^{0}	-3.86×10^{0}	1.11×10^{-6}
F20	-3.32×10^{0}	-3.27×10^{0}	5.99×10^{-2}	-3.32×10^{0}	-3.19×10^{0}	1.06×10^{-1}	-3.32×10^{0}	-3.22×10^{0}	8.66×10^{-2}
F21	-1.02×10^{1}	-8.38×10^{0}	2.59×10^{0}	-1.02×10^{1}	-1.02×10^{1}	5.12×10^{4}	-1.02×10^{1}	-1.02×10^{1}	1.61×10^{-4}
F22	-1.04×10^{1}	-8.66×10^{0}	2.76×10^{0}	-1.04×10^{1}	-1.04×10^{1}	2.32×10^{-4}	-1.04×10^{1}	-1.04×10^{1}	1.69×10^{-4}
F23	-1.05×10^{1}	-9.82×10^{1}	1.86×10^{0}	-1.05×10^{1}	-1.05×10^{1}	1.78×10^{-4}	-1.05×10^{1}	-1.05×10^{1}	3.29×10^{-4}
Win	19	18	21	20	17	20	20	18	17
Draw	2	4	0	2	1	0	2	4	0

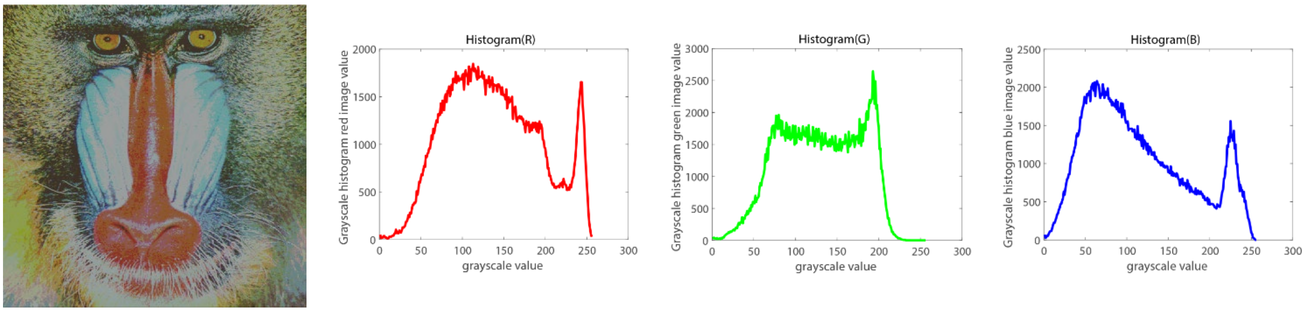
Lose	2	1	2	1	5	3	1	1	6



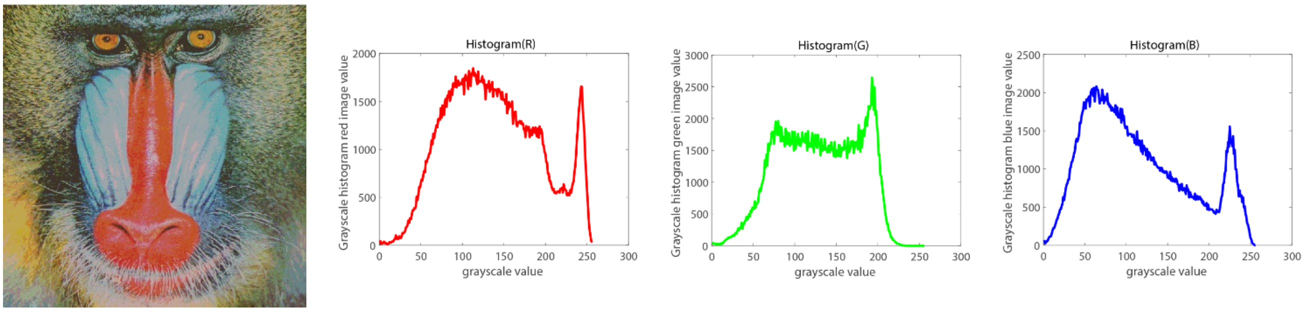
(a) Img1 (b) Img2 (c) Img3 (d) Img4

(e) Img5 (f) Img6 (g) Img7 (h) Img8

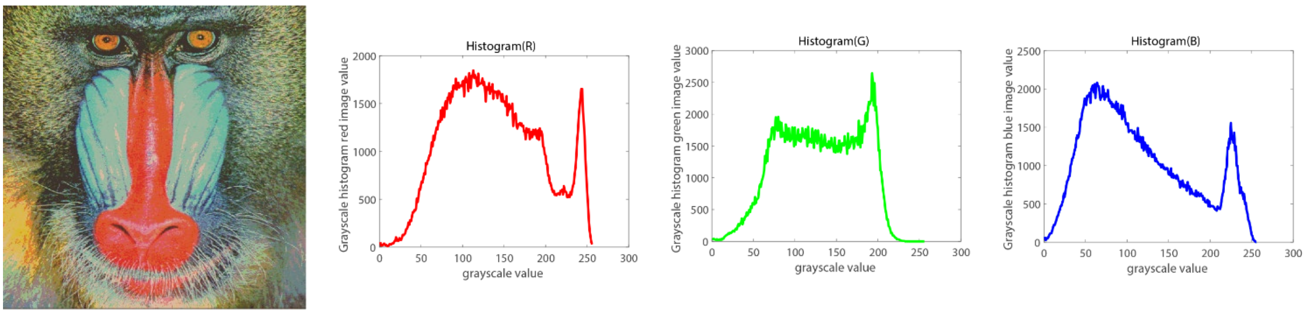
Figure 4. Test color images (Img1- Img8)



(a) 4 threshold (b) Histogram(R) (c) Histogram(G) (d) Histogram(B)



(e) 7 threshold (f) Histogram(R) (g) Histogram(G) (h) Histogram(B)



(i) 10 threshold (j) Histogram(R) (k) Histogram(R) (l) Histogram(R)

Figure 5. The segmented results of Img1 (RGB) based on CDA for 4, 7, 10 threshold values

Figure 6. Segmented image for 4, 7, 10 threshold values (Img2)

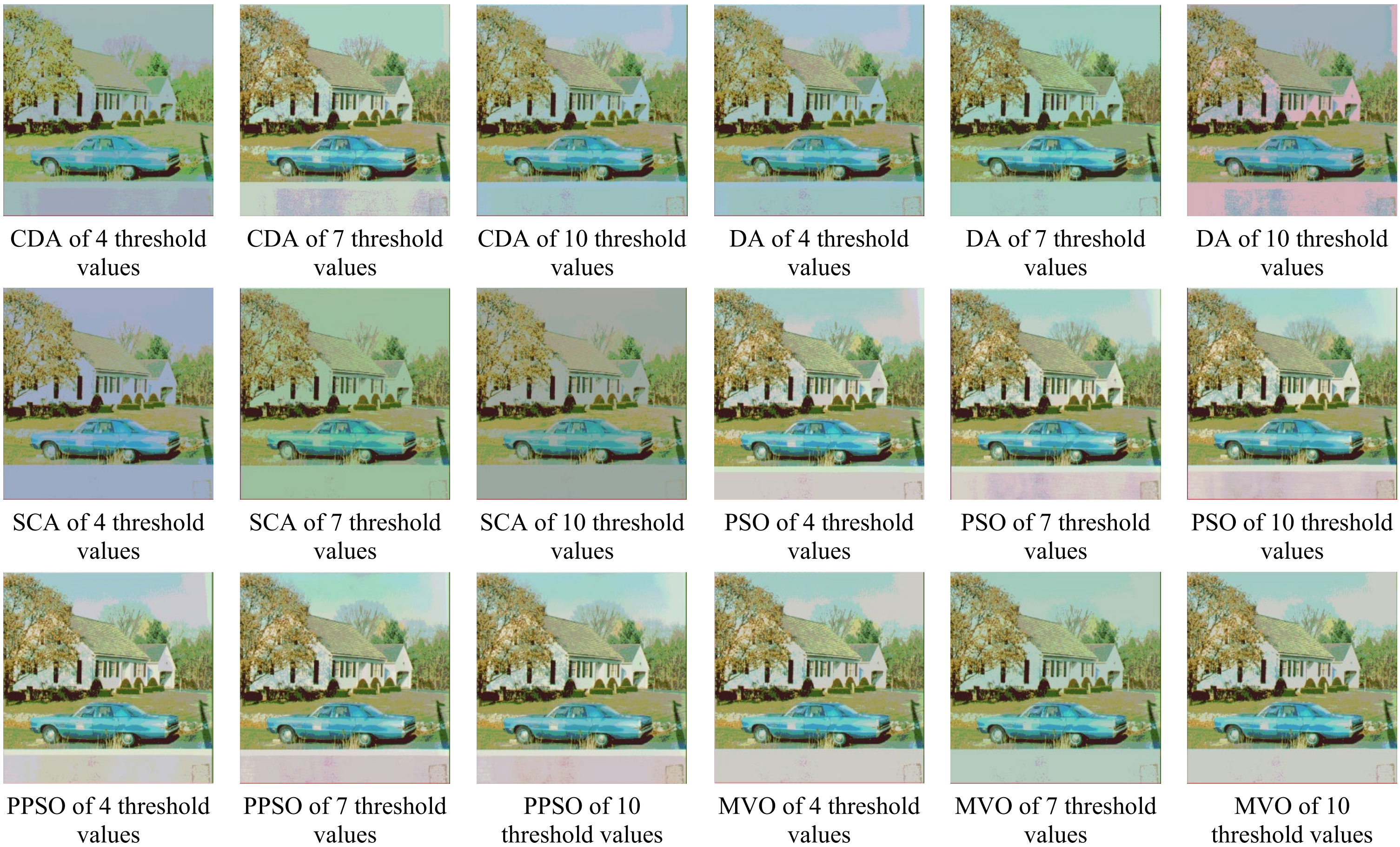


Figure 7. Segmented image for 4, 7, 10 threshold values (Img3)

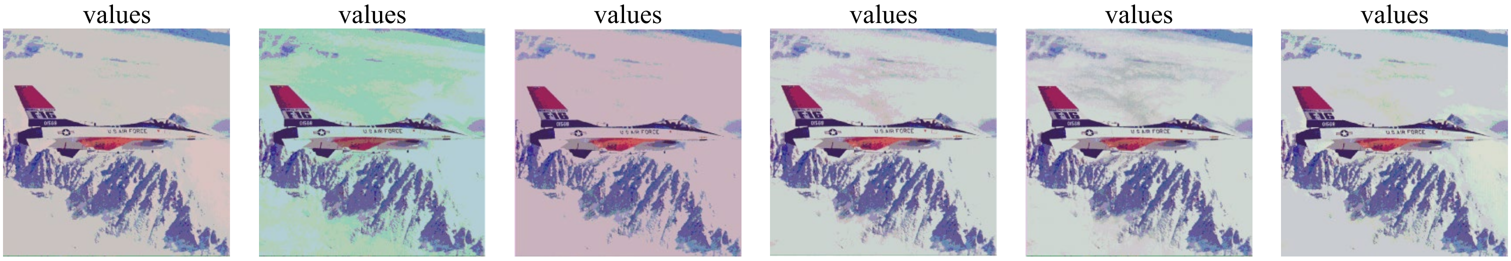
CDA of 4 threshold values | CDA of 7 threshold values | CDA of 10 threshold values | DA of 4 threshold values | DA of 7 threshold values | DA of 10 threshold values

SCA of 4 threshold values | SCA of 7 threshold values | SCA of 10 threshold values | PSO of 4 threshold values | PSO of 7 threshold values | PSO of 10 threshold values

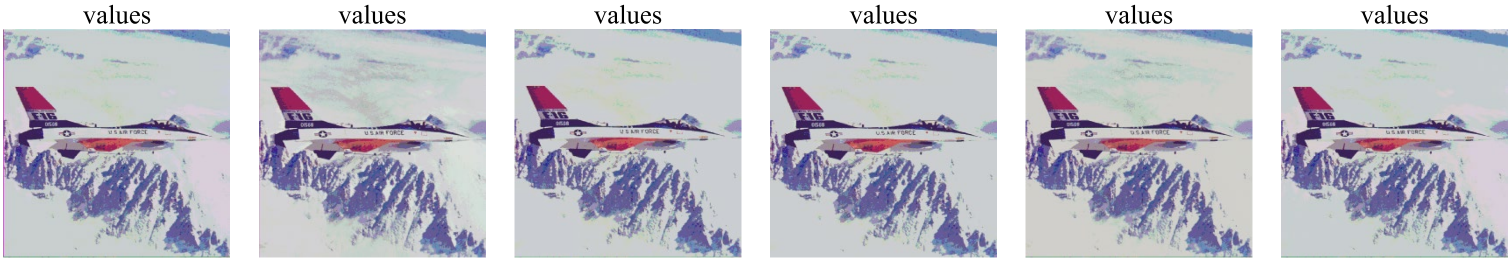
PPSO of 4 threshold values | PPSO of 7 threshold values | PPSO of 10 threshold values | MVO of 4 threshold values | MVO of 7 threshold values | MVO of 10 threshold values

Figure 8. Segmented image for 4, 7, 10 threshold values (Img7)

CDA of 4 threshold values | CDA of 7 threshold values | CDA of 10 threshold values | DA of 4 threshold values | DA of 7 threshold values | DA of 10 threshold values



SCA of 4 threshold values | SCA of 7 threshold values | SCA of 10 threshold values | PSO of 4 threshold values | PSO of 7 threshold values | PSO of 10 threshold values



PPSO of 4 threshold values | PPSO of 7 threshold values | PPSO of 10 threshold values | MVO of 4 threshold values | MVO of 7 threshold values | MVO of 10 threshold values

Figure 9. Segmented image for 4, 7, 10 threshold values (Img8)

Table 8. PSNR parameters calculated by six optimization algorithms on 8 images

Image Name	Threshold	PSNR					
		CDA	DA	SCA	PPSO	PSO	MVO
Image1	4	30.5865	23.4048	21.4763	**30.6363**	26.1008	29.488
	7	**31.1000**	26.8971	20.9546	30.9407	20.6864	22.9031
	10	31.0306	28.1766	21.8964	**31.2875**	25.9995	25.9562
Image2	4	**29.9186**	26.1442	19.0672	29.2423	26.5175	29.1020
	7	**31.3984**	22.1548	18.1388	31.0207	24.4159	23.9879
	10	**31.4821**	20.4787	24.2059	31.2219	25.1399	19.1476
Image3	4	29.8102	24.4413	21.5792	**30.1681**	22.2907	29.0402
	7	31.1891	26.4950	23.4188	**31.2632**	28.6448	27.6871
	10	**31.5293**	23.7732	20.4487	31.2746	24.9039	28.9759
Image4	4	30.1565	20.8929	19.8006	30.8224	**31.7074**	29.4626
	7	31.9784	23.5436	25.8174	**32.1218**	25.9616	26.3943
	10	**32.4730**	22.1709	25.7433	32.2489	27.9548	30.2464
Image5	4	33.5849	33.2467	27.1658	**35.1218**	32.6137	34.5553
	7	34.6069	31.7129	27.3821	34.7597	32.3667	32.2654
	10	**36.2270**	30.3762	26.4091	36.0267	29.7949	32.2696
Image6	4	**30.4792**	18.7616	18.7616	29.3889	23.3636	28.8609
	7	31.5053	24.0224	22.8516	31.6592	26.8754	26.6322
	10	31.4836	17.7118	16.3324	30.6472	22.9569	25.9651
Image7	4	**32.1163**	22.2008	20.1449	32.0903	30.5150	31.9067
	7	32.4726	28.7798	19.8580	**33.3446**	29.4493	33.1932
	10	31.0225	22.9511	19.4995	**32.0617**	23.5313	23.6609
Image8	4	**31.3416**	24.4935	27.8479	30.1746	28.9595	29.4314
	7	32.5696	26.2809	27.3677	**32.9877**	24.8387	30.9894
	10	**30.6790**	28.6486	24.1023	29.6723	24.5005	30.2912

Table 9. SSIM parameters calculated by six optimization algorithms on 8 images

Image Name	Threshold	SSIM					
		CDA	DA	SCA	PPSO	PSO	MVO
Image1	4	**0.9018**	0.7465	0.6413	0.8991	0.8014	0.8890
	7	0.9094	0.8433	0.7002	**0.9097**	0.6069	0.8156
	10	0.9052	0.8128	0.7680	**0.9116**	0.6524	0.8570
Image2	4	**0.9437**	0.9028	0.7562	0.9356	0.8999	0.9259
	7	**0.952**3	0.8685	0.7389	0.9498	0.8949	0.8639
	10	**0.9533**	0.7886	0.8687	0.9532	0.8966	0.7586
Image3	4	**0.8394**	0.7191	0.5904	0.8259	0.6644	0.8105
	7	**0.8903**	0.6788	0.6724	0.8849	0.7807	0.7425
	10	**0.9064**	0.7533	0.6523	0.8951	0.6382	0.8059
Image4	4	0.7987	0.7169	0.6489	0.8067	0.8033	**0.8093**
	7	**0.8364**	0.7205	0.6872	0.8363	0.7486	0.7872
	10	**0.8394**	0.6886	0.6679	0.8380	0.7370	0.8103
Image5	4	**0.9614**	0.8600	0.8820	0.9599	0.9526	0.9584
	7	0.9568	0.9145	0.8998	**0.9664**	0.9521	0.9589
	10	0.9633	0.8811	0.8368	**0.9638**	0.9122	0.9345
Image6	4	**0.9072**	0.5645	0.6085	0.8872	0.7893	0.8702
	7	0.9154	0.2998	0.8190	**0.9276**	0.7678	0.8129
	10	**0.9196**	0.5535	0.4263	0.8995	0.6314	0.7610
Image7	4	0.9315	0.8074	0.4575	**0.9319**	0.9183	0.9296
	7	0.9328	0.8992	0.4895	**0.9380**	0.9048	0.9331
	10	0.9279	0.8365	0.4538	**0.9334**	0.8297	0.8289
Image8	4	**0.8493**	0.5857	0.6579	0.8192	0.7223	0.8489
	7	**0.8860**	0.5796	0.5497	0.8847	0.5142	0.8337
	10	0.8278	0.8206	0.4669	**0.8304**	0.5239	0.8199

Table 10. FSIM parameters calculated by six optimization algorithms on 8 images

Image Name	Threshold	FSIM					
		CDA	DA	SCA	PPSO	PSO	MVO
Image1	4	0.9636	0.8918	0.8388	**0.9645**	0.9243	0.9510
	7	**0.9677**	0.9356	0.8491	0.9662	0.8347	0.8666
	10	0.9666	0.9392	0.8652	**0.9690**	0.9145	0.9195
Image2	4	**0.9176**	0.8845	0.7994	0.9093	0.8724	0.8978
	7	**0.9397**	0.8394	0.7993	0.9329	0.8779	0.8821
	10	**0.9417**	0.8485	0.8454	0.9402	0.8811	0.8172
Image3	4	0.9328	0.8893	0.8376	**0.9367**	0.8440	0.9178
	7	**0.9507**	0.8837	0.8608	0.9478	0.9233	0.8996
	10	**0.9521**	0.8910	0.8252	0.9496	0.8718	0.9167
Image4	4	0.8340	0.7438	0.7692	0.8404	**0.8559**	0.8314
	7	0.8623	0.8020	0.7997	**0.8694**	0.8013	0.8146
	10	**0.8751**	0.7722	0.8058	0.8676	0.8418	0.8412
Image5	4	0.9243	0.9121	0.9033	0.9252	0.9153	**0.9324**
	7	**0.9326**	0.9015	0.9012	0.9295	0.9154	0.9170
	10	**0.9403**	0.9017	0.9203	0.9339	0.8891	0.9194
Image6	4	**0.9514**	0.8623	0.8355	0.9485	0.9064	0.9421
	7	**0.964**3	0.8720	0.9153	0.9635	0.9291	0.9261
	10	**0.9636**	0.8392	0.8207	0.9581	0.9104	0.9298
Image7	4	**0.8763**	0.7812	0.7988	0.8748	0.8562	0.8633
	7	0.8770	0.8546	0.7726	**0.8933**	0.8442	0.8844
	10	0.8516	0.8148	0.7707	**0.8736**	0.8206	0.8033
Image8	4	**0.9313**	0.8876	0.9000	0.9130	0.9165	0.9029
	7	0.9445	0.9086	0.9087	**0.9466**	0.9048	0.9255
	10	**0.9169**	0.8953	0.8766	0.9046	0.8979	0.9153

Table 11. MS-SSIM parameters calculated by six optimization algorithms on 8 images

Image Name	Threshold	MS-SSIM					
		CDA	DA	SCA	PPSO	PSO	MVO
Image1	4	**0.9754**	0.9021	0.8071	0.9748	0.9449	0.9640
	7	**0.9775**	0.9470	0.8218	0.9767	0.7749	0.8616
	10	0.9762	0.9556	0.8715	**0.9787**	0.9393	0.9337
Image2	4	**0.9416**	0.9229	0.8128	0.9360	0.9119	0.9349
	7	**0.9582**	0.8643	0.8068	0.9528	0.9102	0.9109
	10	**0.960**1	0.8683	0.8682	0.9564	0.9105	0.8257
Image3	4	0.9602	0.9223	0.8715	**0.9641**	0.8746	0.9442
	7	**0.9718**	0.9342	0.8998	0.9696	0.9465	0.9385
	10	**0.9760**	0.9074	0.8427	0.9712	0.9151	0.9440
Image4	4	0.9445	0.7730	0.6950	0.9464	**0.9578**	0.9437
	7	0.9503	0.8697	0.8904	**0.9515**	0.8934	0.9046
	10	**0.9518**	0.8322	0.8805	0.9504	0.9385	0.9452
Image5	4	0.9825	0.9828	0.9744	0.9834	0.9821	**0.9850**
	7	**0.9855**	0.9802	0.9431	0.9853	0.9819	0.9808
	10	**0.9915**	0.9783	0.9746	0.9881	0.9726	0.9816
Image6	4	**0.9609**	0.8816	0.7926	0.9524	0.9145	0.9556
	7	0.9732	0.9056	0.8659	**0.9755**	0.9401	0.9478
	10	**0.9729**	0.8143	0.7643	0.9672	0.9026	0.9422
Image7	4	**0.9651**	0.8709	0.8415	0.9649	0.9525	0.9634
	7	0.9630	0.9436	0.7847	**0.9672**	0.9462	0.9671
	10	0.9600	0.8874	0.7807	**0.9640**	0.9053	0.8919
Image8	4	**0.9626**	0.9184	0.9315	0.9410	0.9548	0.9344
	7	0.9744	0.9440	0.9520	**0.9783**	0.9509	0.9612
	10	**0.9445**	0.9264	0.9131	0.9409	0.9403	0.9417

5 Conclusion

In this paper, a collaborative evolutionary dragonfly algorithm is proposed, which makes full use of the multi-group mechanism to strengthen the cooperation and communication between groups. CDA compared with other algorithms in 23 test functions, experimental results prove that the proposed CDA algorithm is better than other algorithms. After analyzing the mechanism of swarm intelligence algorithm and pattern search algorithm, the improved algorithm is applied to the field of multi-threshold segmentation. With the increase in the number of thresholds, traditional threshold processing methods can no longer meet the requirements of real-time applications. The data results show that it is more competitive than the original algorithm. Then from experimental results of the multi-threshold color image segmentation show that when different threshold standards are used, the performance of swarm intelligence algorithms is different. After applying the proposed strategy, the CDA algorithm not only has good global exploration and local exploitation, but also has superior performance in multi-threshold color image segmentation based on the minimum cross-entropy method.

References

[1] S.-C. Chu, X. Xue, J.-S. Pan, X. Wu, Optimizing ontology alignment in vector space, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 15-22, January, 2020.

[2] S.-C. Chu, H.-C. Huang, J. F. Roddick, J.-S. Pan, Overview of algorithms for swarm intelligence, *International Conference on Computational Collective Intelligence*, Springer, Vol. 6922, pp. 28-41, September, 2011.

[3] D. E. Goldberg, J. H. Holland, *Genetic algorithms and machine learning*, 1988.

[4] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snášel, A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111- 1118, July, 2018.

[5] T. R. Farshi, J. H. Drake, E. Özcan, A multimodal particle swarm optimization-based approach for image segmentation, *Expert Systems with Applications*, Vol. 149, pp. 113233, July, 2020.

[6] S.-C. Chu, P.-W. Tsai, J.-S. Pan, Cat swarm optimization, *Pacific Rim international conference on artificial intelligence*, Springer, Vol. 4099, pp. 854-858, August, 2006.

[7] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, Vol. 11, No. 4, pp. 341-359, December, 1997.

[8] J.-S. Pan, Z. Meng, H. Xu, X. Li, A matrix-based implementation of DE algorithm: the compensation and deficiency, *International Conference on Industrial, Engineering and Other Applications of Applied*

Intelligent Systems, Springer, Vol. 10350, pp. 72-81, June, 2017.

[9] S.-C. Chu, J. F. Roddick, C.-J. Su, J.-S. Pan, Constrained ant colony optimization for data clustering, *Pacific Rim International Conference on Artificial Intelligence*, Springer, Vol. 3157, pp. 534-543, August, 2004.

[10] D. Karaboga, B. Basturk, On the performance of artificial bee colony (abc) algorithm, *Applied soft computing*, Vol. 8, No. 1, pp. 687-697, January, 2008.

[11] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization*, Vol. 39, pp. 459-471, April, 2007.

[12] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in engineering software*, Vol. 69, pp. 46-61, March, 2014.

[13] P. Hu, J.-S. Pan, S.-C. Chu, Improved binary grey wolf optimizer and its application for feature selection, *Knowledge-Based Systems*, Vol. 195, pp. 105746, May, 2020.

[14] J.-S. Pan, P. Hu, S.-C. Chu, Novel parallel heterogeneous meta- heuristic and its communication strategies for the prediction of wind power, *Processes*, Vol. 7, No. 11, pp. 845, November, 2019.

[15] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intel- ligence optimizer for air robot path planning, *International journal of intelligent computing and cybernetics*, Vol. 7, No. 1, pp. 14-37, March, 2014.

[16] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, W.-M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, Vol. 12, No. 3, pp. 767, January, 2020.

[17] X.-S. Yang, S. Deb, Cuckoo search via lévy flights, *2009 World congress on nature & biologically inspired computing (NaBIC). IEEE*, pp. 210-214, December, 2009.

[18] J.-S. Pan, P.-C. Song, S.-C. Chu, Y.-J. Peng, Improved compact cuckoo search algorithm applied to location of drone logistics hub, *Mathematics*, Vol. 8, No. 3, pp. 333, March, 2020.

[19] P.-C. Song, J.-S. Pan, S.-C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, *Applied Soft Computing*, Vol. 94, pp. 106443, September, 2020.

[20] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, P. Hu, Quasi-affine transformation evolutionary algorithm with communication schemes for application of rssi in wireless sensor networks, *IEEE Access*, Vol. 8, pp. 8583–8594, January, 2020.

[21] Z. Meng, J.-S. Pan, Quasi-affine transformation evolutionary (quatre) algorithm: A parameter-reduced differential evolution algorithm for optimization problems, *2016 IEEE Congress on Evolutionary Computation (CEC). IEEE*, pp. 4082-4089, July, 2016.

[22] N. Liu, J.-S. Pan, C. Sun, S.-C. Chu, An efficient surrogate-assisted quasi-affine transformation evolutionary algorithm for expensive optimization problems, *Knowledge-Based Systems*, Vol. 209, pp. 106418, December, 2020.

[23] S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, Vol. 27, No. 2, pp. 495-513, March, 2016.

[24] G. I. Sayed, A. Darwish, A. E. Hassanien, Quantum multiverse optimization algorithm for optimization problems, *Neural Computing and Applications*, Vol. 31, pp. 2763-2780, November, 2019.

[25] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization tech- nique for solving single-objective, discrete, and multi-objective problems, *Neural Computing and Applications*, Vol. 27, No. 4, pp. 1053-1073, May, 2016.

[26] P. T. Daely, S. Y. Shin, Range based wireless node localization using dragonfly algorithm, *2016 eighth international conference on ubiquitous and future networks (ICUFN). IEEE*, pp. 1012-1015, August, 2016.

[27] D. Das, A. Bhattacharya, R. N. Ray, Dragonfly algorithm for solving probabilistic economic load dispatch problems, *Neural Computing and Applications*, Vol. 32, No. 8, pp. 3029-3045, May, 2020.

[28] M. Xu, Y.-P. Feng, Z.-M. Lu, Fast feature extraction based on multi- feature classification for color image, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 2, March, 2019.

[29] J.-S. Pan, X. Wang, S.-C. Chu, T. Nguyen, A multi-group grasshop- per optimisation algorithm for application in capacitated vehicle routing problem, *Data Science and Pattern Recognition*, Vol. 4, No. 1, pp. 41-56, July, 2020.

[30] W. Song, N. Zheng, R. Zheng, X.-B. Zhao, A. Wang, Digital image semantic segmentation algorithms: A survey, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 196-211, January, 2019.

[31] C. Du, S. Gao, Image segmentation-based multi-focus image fusion through multi-scale convolutional neural network, *IEEE access*, Vol. 5, pp. 15750-15 761, August, 2017.

[32] S. Yin, Y. Zhang, S. Karim, Large scale remote sensing image segmentation based on fuzzy region competition and gaussian mixture model, *IEEE Access*, Vol. 6, pp. 26069-26080, May, 2018.

[33] H. Jia, X. Peng, W. Song, C. Lang, Z. Xing, K. Sun, Multiverse optimization algorithm based on lévy flight improvement for multithreshold color image segmentation, *IEEE Access*, Vol. 7, pp. 32 805-32 844, March, 2019.

[34] H. Liang, H. Jia, Z. Xing, J. Ma, X. Peng, Modified grasshopper algorithm-based multilevel thresholding for color image segmentation, *IEEE Access*, Vol. 7, pp. 11 258-11 295, January, 2019.

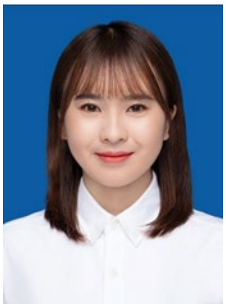
[35] X. Li, J. Zhang, M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, *Neural Computing and Applications*, Vol. 24, No. 7-8, pp. 1867-1877, June, 2014.

[36] Q. W. Chai, S. C. Chu, J. S. Pan, W. M. Zheng, Applying Adaptive and Self Assessment Fish Migration Optimization on Localization of Wireless Sensor Network on 3-D Terrain, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 11, No. 2, pp. 90-102, June, 2020.

[37] J. Korhonen, J. You, Peak signal-to-noise ratio revisited: Is simple beautiful, *2012 Fourth International*

Workshop on Quality of Multime- dia Experience. IEEE, pp. 37-38, August, 2012.
[38] L. Zhang, L. Zhang, X. Mou, D. Zhang, Fsim: A feature similarity index for image quality assessment, *IEEE transactions on Image Processing*, Vol. 20, No. 8, pp. 2378-2386, January, 2011.
[39] Z. Wang, Q. Li, Information content weighting for perceptual image quality assessment, *IEEE Transactions on image processing*, Vol. 20, No. 5, pp. 1185–1198, November, 2010.

Biographies



Fei-Fei Liu received her B.S. degree from Shengli College China University Of Petroleum in 2020. She is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests are swarm intelligence and image processing.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She is the Research Fel- low in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, She is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.

Xiaopeng Wang rreceived the B.S. degree in electrical engineering from the Fujian University of Technology, China, in 2016, the M.S. degree in electrical engineering from the Fujian University of Technology, China, in 2020. He is currently working in currently working Shandong University of Science and Technology. His current research interests include pattern recognition and swarm intelligence.

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 the Director of the Fujian Provincial Key Lab of Big Data Mining and Applications, and an Assistant President with the Fujian University of Technology. He is also the Professor with the Harbin Institute of Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. He was offered Thousand Talent Program in China in 2010.