Article

Not peer-reviewed version

A Fast Multi-Threshold Color Image Segmentation Algorithm Using Bayesian Forecasting Evolutionary Algorithm

Yunzhi Jiang , Wei-chang Yeh * , Zhengchun Lin *

Posted Date: 26 June 2024

doi: 10.20944/preprints202406.1791.v1

Keywords: Bayesian Forecasting Evolutionary Algorithm (BFEA); Multilevel Thresholding Technique; Color Image Segmentation; Prediction Vector

Article

A Fast Multi-Threshold Color Image Segmentation Algorithm Using Bayesian Forecasting Evolutionary Algorithm

Yunzhi Jiang [1], Wei-chang Yeh [2,*] and Zhengchun Lin [1,*]

[1] School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China
[2] IEEM, National Tsing Hua University, Hsinchu, Taiwan
* Correspondence: yeh@ieee.org, eric.green@qq.com

Abstract: Multi-threshold image segmentation is a relevant research field in further higher-level image preprocessing and computer vision. This paper adopts the Bayesian Forecasting Evolutionary Algorithm (BFEA) for natural scenery image segmentation using multilevel thresholding. An update method for the prediction vector is designed for image segmentation. The design idea of the BFEA algorithm for multilevel thresholding is based on the basic principle of Evolutionary Computation, the probability distribution of promising solutions, and the Bayesian theorem. BFEA could effectively solve the curse of dimensionality. Extensive experiments have demonstrated that the proposed algorithm dominated the state-of-the-art population-based thresholding method in term of the quality of image, function fitness value, optimal threshold value, SSIM value, PSNR value, computation time, dimensionality issue and convergence rate. The proposed method is very effective for color image segmentation.

Keywords: bayesian forecasting evolutionary algorithm (BFEA); multilevel thresholding technique; color image segmentation; prediction vector

I. INTRODUCTION

Image and video are the most important carriers and media of information communication in people's life. Wireless Multimedia Sensor Networks (WMSN) differs from wireless sensor networks and wired networks. There are some problems to be solved in this area, such as the size of transmitted data, memory resources, and power consumption of each node, and so on. Image compression technology is one of the most effective solutions to overcome those problems. Due to the redundancy caused by the high correlation between a very large number of pixels in the image or video, many compression algorithms are developed. Multilevel image thresholding technology is widely used in image preprocessing or video compression, and it is a challenging task. However, due to the increase in the number of thresholds in multilevel image thresholding, the computational complexity increases exponentially [1–3]. In order to solve the problem, many multi-level threshold methods based on Intelligent Optimization Algorithm (IOA) have been proposed in recent years.

The Electromagnetism-like Algorithm for multilevel thresholding was introduced by Oliva et al. [2]. The algorithm can effectively calculate the threshold values within the reduced number of generations. An improved Particle Swarm Optimization Algorithm based on quantum behavior (CQPSO) was proposed by Gao et al. [4]. The algorithm uses a collaborative method to improve efficiency. Dey et al. [5] also concluded that the Particle Swarm Optimization based on quantum-inspired was superior to the other five methods using quantum meta-heuristic. Mohamed Abd El et al. [6] proposed two new swarm algorithms to determine the optimal multilevel threshold for image segmentation. One is the Whale Optimization Algorithm (WOA), which imitates the natural cooperative behavior of whales; the other is the Moth-Flame Optimization (MFO), which mimics the

 doi:10.20944/preprints202406.1791.v1

natural behavior of moths.These solutions were assessed using Otsu's fitness functions. MFO, which struck a better balance between exploitation and exploration, showed better results than WOA. Kumar et al. [7] proposed a different objective function that uses the homogeneity histogram of the image and the original histogram for image segmentation to improve the efficiency of the evolutionary algorithm. Zhao et al. [8] proposed a multi-objective evolutionary algorithm to perform image segmentation tasks based on intuitionistic fuzzy clustering and image spatial information. M.S.R. Naidu et al. proposed a multilevel image thresholding method based on the inspiration of fireflies by using fuzzy entropy or maximizing Shannon entropy [9]. Experiments showed that using the maximum Fuzzy entropy or Shannon entropy as the fitness function in the firefly algorithm is a very efficient image thresholding method. A multi-objective clustering-based PSO image segmentation method was proposed by Liu et al. [10], in which a similarity computing method based on electromagnetic forces was offered, and the appropriate cluster number was obtained according to the scale-space theory. Bhandari et al. [11] proposed the 3D Otsu's based Cuttlefish Algorithm (CFA) for color image multilevel thresholding segmentation to get optimal threshold values. Experimental results showed that, compared with other algorithms listed in the literature, 3D Otsu's based-CFA had obtained the best results. In the paper [12], the three strategies, "the Gaussian mutation, the Levy flight, and the opposition-based learning," were employed in Emperor Penguin Optimization (EPO) to increase the searchability, jump out local optimization.

The Bee Mating Algorithm (HBMA) is an important intelligent optimization algorithm that simulates the mating process of real bees. Horng [13] proposed a honey bee mating optimization algorithm (MEHBMOT) for image segmentation, which obtained multilevel threshold selection based on maximum entropy criterion. Jiang et al. [1] proposed an improved honey bee mating optimization by using the cooperative strategy (CHBMA). However, the importance of dynamically generating a high-quality population in each generation was neglected in MEHBMOT and CHBMA algorithms. Hence, a fast multi-threshold color image segmentation algorithm by using Bayesian Forecasting Evolutionary Algorithm [14] is proposed in this paper, in which the prediction vector $PVector[n]$ is real-time dynamically determined based on Bayesian forecasting technology, and new population can be dynamically generated according to $PVector[n]$. Such a strategy can increase computational efficiency and reduce the complexity of multi-level threshold processing.

The main contributions of this paper are the following four points. First, this paper established a Bayesian Forecasting Evolutionary Algorithm (BFEA) based multilevel thresholding for color image segmentation after BFEA [14] was proposed for application in function optimization. Second, BFEA was often used to solve continuous function optimization before. However, it is simplified and applied in the discrete combinatorial optimization problem that is color image segmentation in this paper. Third, combining a population initialization strategy with BFEA can jump out of local optimum as much as possible. Fourth, an update method for the prediction vector is designed for image segmentation.

Section II describes related work. Section III presents BFEA and its analysis. Section IV presents performance evaluation and experimental study. Finally, some conclusions are drawn in Section V.

II. RELATED WORK

A. Bayesian Forecasting Evolutionary Algorithm

Bayesian Forecasting Evolutionary Algorithm was proposed for solving the linkage problem and deceptive problem effectively. In this paper, it is employed in image thresholding for solving the curse of dimensionality in image segmentation.

BFEA combines Bayesian theorem with the probability distribution over the set of promising solutions and integrates with the basic principle of evolutionary computation. In BFEA, it guides the exploration of the search space according to the prediction vector $PVector[n]$, each component of which is the corresponding subspace's prediction probability including the optimal solutions. Both the information from all the generated generations and the prior information about the problem are used to estimate the prediction vector. This algorithm is also proved to converge fast to the subspaces with optimal solutions at a faster speed [14]. The procedure of BFEA is shown in Figure 1.

```
Initialize the prediction vector with uniform distribution;
Initialize the population;
WHILE not stop
{
    Introduce a new population;
    Choose a prediction method to build a prediction model;
    Predict the probability distribution of the promising solution set;
    Update the prediction vector;
    Select some promising individuals from the population;
    Perform other operations such as mutation;
    Evaluate individuals;
} END WHILE
```

Figure 1. Procedure of BFEA.

B. Measure based on entropy criterion

Let L be gray levels and N pixels in an image and gray levels are ranged over [0, L-1]. Let h(i) be the observed frequency of gray-level i; let

$$N=h(0)+\ldots+h(i)+\ldots+h(L-1) \tag{1}$$

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

Suppose there are M thresholds: $\{t_1, t_2,\ldots,t_M\}$, where ($1\leq M\leq L-1$), these thresholds divide the original image into M+1 clusters, represented by the following notation:

$C_0= \{0,1,\ldots,t_1-1\}$,

$C_1= \{t_1, t_1+2,\ldots, t_2-1\}$,

…,

$C_M = \{t_M, t_M+2,\ldots,L-1\}$.

Kapur entropy was initially proposed to deal with the double threshold image segmentation problem, and we can extend it to the multilevel threshold image segmentation as given by Equation (3). We maximize the equation to search these optimal thresholds: $\{t_1, t_2, \ldots, t_M\}$. The proposed algorithm in this paper uses the formula (3) as the objective function.

$$\begin{gathered}
f(t_1,t_2,\ldots,t_M)=E_0+E_1+E_2+\cdots E_M \\
\omega_0=\sum_{i=0}^{t_1-1}p_i,\ E_0=-\sum_{i=0}^{t_1-1}\frac{p_i}{\omega_0}\ln\frac{p_i}{\omega_0} \\
\omega_1=\sum_{i=t_1}^{t_2-1}p_i,\ E_1=-\sum_{i=t_1}^{t_2-1}\frac{p_i}{\omega_1}\ln\frac{p_i}{\omega_1} \\
\omega_2=\sum_{i=t_2}^{t_3-1}p_i,\ E_2=-\sum_{i=t_2}^{t_3-1}\frac{p_i}{\omega_2}\ln\frac{p_i}{\omega_2} \\
\vdots \\
\omega_M=\sum_{i=t_M}^{L-1}p_i,\ E_M=-\sum_{i=t_M}^{L-1}\frac{p_i}{\omega_M}\ln\frac{p_i}{\omega_M}
\end{gathered} \tag{3}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

III. THE PROCEDURE AND ANALYSIS OF THE PROPOSED ALGORITHM

In this section, it is explained how Bayesian Forecasting Evolutionary Algorithm using Kapur's entropy as the objective function is applied to the fast multi-threshold image segmentation problem. The flow chart is shown in Figure 2. Next, we introduce the detailed process of BFEA for color image segmentation and some analysis of it.

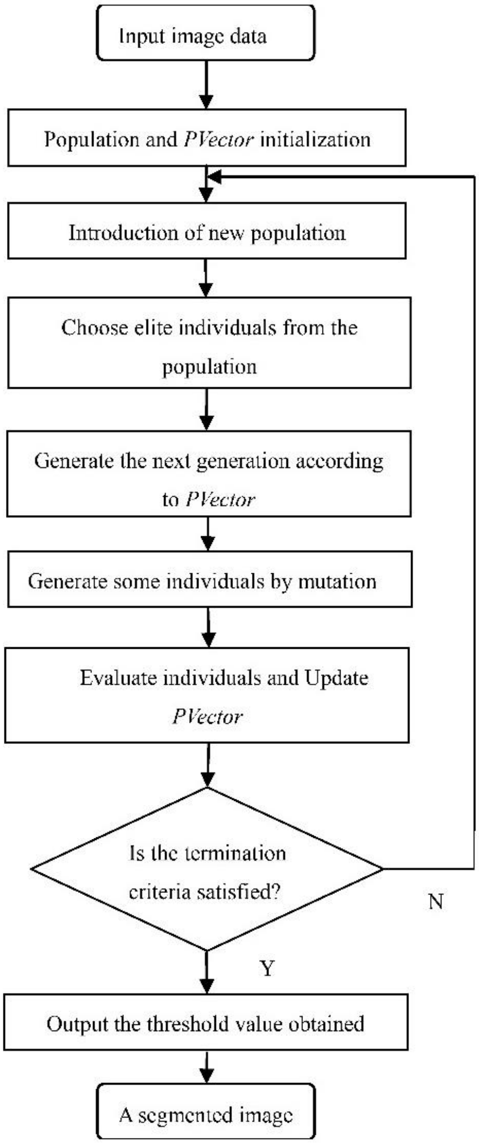


Figure 2. The flowchart of BFEA for color image segmentation.

In Section 3.A, we designed the initialization strategy of BFEA. In Section 3.B, we explained the reason for adding one new population in each generation. In Section 3.C, we explained the prediction method of the proposed algorithm. In Section 3.D, we designed the breeding operator. In Section 3.E, we explained the mutation operator. In Section 3.F, we give reasons and rules for deleting similar thresholds. In Section 3.G, the parameter settings used in the proposed algorithm are given.

A. The initialization scheme

The population is initialized as follows. We can generate the position of an arbitrary individual $\mathbf{X}_i$ in d^{th} dimension, i.e. $\mathbf{X}_{id}$, from in the closed interval [(d-1)×floor(L/D), d×floor(L/D)] (i=1, 2, …, PopulationSize) according to the uniform distribution, with using the property of image multilevel thresholds arranged from a low gray level to a high one. Among all individuals, the one with the maximum objective function value is selected as the best individual **BestX**.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

B. The Introduction of New population

We imitate the natural phenomenon that a homogeneous swarm in nature needs to continually interact with new homogenous swarm to prevent the population from generating too many eccentric and homogeneous individuals in BFEA. Considering the efficiency, we adopted the strategy of reducing the initial population size and adding a new population to each generation. The size of new populations is the same as the size of old populations.

C. The proposed prediction method

We select some elites from the initial population. The new population is generated by elite individuals chosen from the current generation in the BFEA search space. *TopSize* represents the number of elites, and *TopSize* is set to 10. This strategy can increase the population diversity, and the overall searchability can also be significantly improved.

Therefore, the difference between the j^{th} individual X_j in the elite individuals and the current best individual *BestX* is denoted by the vector ($\boldsymbol{X_j}$-***BestX***). How many individuals should one elite individual generate? It's decided by the corresponding component of the prediction vector ***PVector***[1, 2, …, *TopSize*]. The j^{th} elite individual will generate *PopulationSize* ×***PVector***[*j*] (*j*=1, 2, …, *TopSize*) individuals. How to initialize and update the prediction vector is discussed below.

Each component of the prediction vector is initialized with (1/TopSize) in the first generation of the population (t=0). That is, the prediction vector is initialized with uniform distribution when the population is initialized. For the (t+1) generations, the components of the prediction vector should be assigned separately so that they have different fertility, according to varying levels of excellence. With the creation of the new generation population, the method can adjust **PVector**[j] dynamically according to the difference vector ($\mathbf{X}_j$-**BestX**) and the corresponding posterior probability in BFEA Algorithm. **PVector**[j] varies with the corresponding the distance ($\mathbf{X}_j$-**BestX**). If an elite individual j^{th} is equal to the **BestX**, the **PVector**[j] will be the largest one of the prediction vector. Then BFEA Algorithm updates it according to the following expression Equation (4).

$$P(A_j^{(t+1)}) = P(A_j^{(t)} \mid B) = \frac{P(A_j^{(t)}B)}{P(B)} = \frac{P(A_j^{(t)})P(B|A_j^{(t)})}{\sum_{j=1}^{TopSize} P(A_j^{(t)})P(B|A_j^{(t)})} \tag{4}$$

Regarding $P(B \mid A_j)$, different calculation methods can be used for different types of problems. A different update method for the prediction vector is designed for image segmentation. In our experiment, it was found that there was not much difference in the fitness value of excellent individuals. If the probability $P(B \mid A_j)$ was directly calculated based on the fitness value, the effect would be not good. Therefore, the following expression Equation (5) is applied to calculate $P(B \mid A_j)$ in the proposed algorithm.

$$P(B \mid A_j) = \frac{(f(BestX) - f(X_j))}{\sum_{i=1}^{TopSize} (f(BestX) - f(X_j))} \tag{5}$$

The MATLAB Pseudocode for the Bayesian prediction method is shown in Figure 3. In order to prevent the premature convergence of the algorithm, we need to construct an operator, doubly stochastic matrix operator (denoted by ***P***). The operator is shown in Equation (6), assuming that *s*=1+2+…+*TopSize*, whose function is theoretically explained in the paper [15]. The experimental analysis of the effect of this matrix operator on the convergence rate is shown in Figure 4. The BFEA parameters are shown in Table 1(a), except that the number of threshold value is set to 5. In this experiment, we tested the corresponding grayscale images of color images (Test01- Test08). Each experiment simulated 25 runs, and we used the average of the results.

$$\mathbf{P}=\begin{pmatrix} 1/s & 2/s \cdots & TopSize/s \\ 2/s & 3/s \cdots & 1/s \\ \vdots & \vdots & \vdots \\ TopSize/s & 1/s \cdots & (TopSize-1)/s \end{pmatrix} \tag{6}$$

% ***PVector*** represents the prediction vector

% *t* denotes the t^{th} generation

% *BestXfitness* is the current best fitness value

% ***EliteFitness*** is a collection of fitness values for the elites selected.

% ***P*** is the doubly stochastic matrix

FitnessBayes=*BestXfitness*-***EliteFitness***;

FitnessBayes=***FitnessBayes***./sum(***FitnessBayes***);

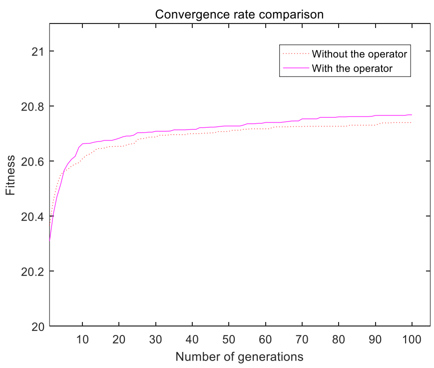
PVector=***FitnessBayes***.****PVector***./sum(***FitnessBayes***.****PVector***);

IFmod(*t*,5)==0

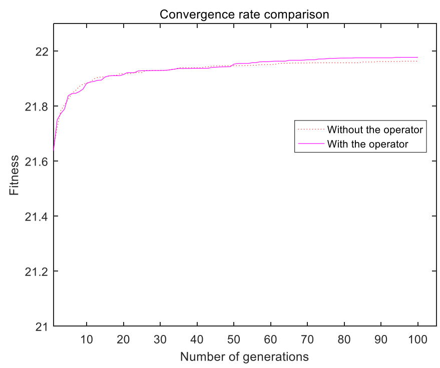
PVector=***PVector*******P***;

END

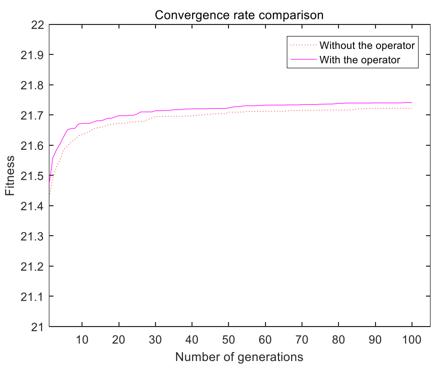
Figure 3. The MATLAB Pseudocode of Bayesian prediction method.



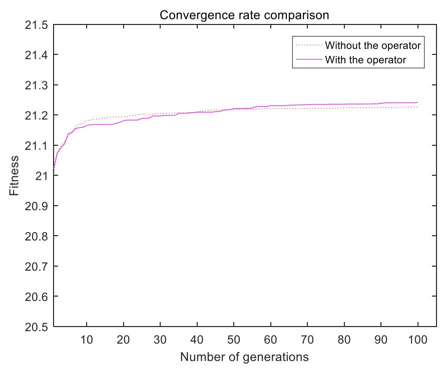
(a) Fitness values in successive generations - image Test01



(b) Fitness values in successive generations - image Test02

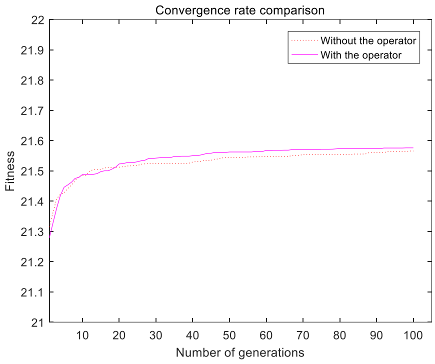


(c) Fitness values in successive generations - image Test03

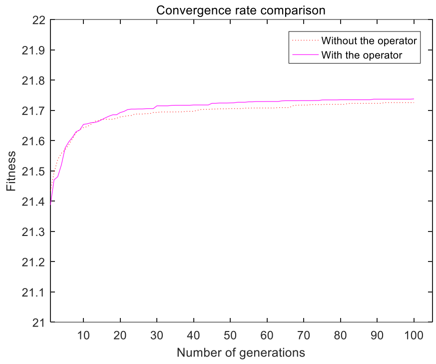


(d) Fitness values in successive generations - image Test04

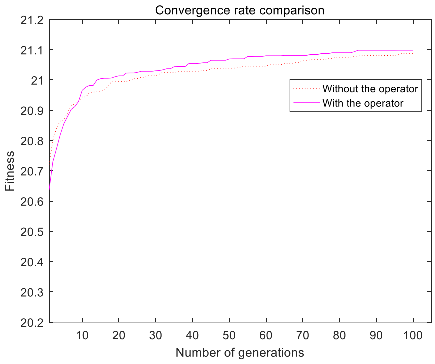
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1



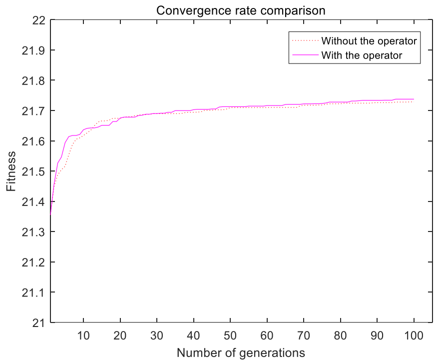
(e) Fitness values in successive generations - image Test05



(f) Fitness values in successive generations - image Test06



(g) Fitness values in successive generations - image Test07



(h) Fitness values in successive generations - image Test08

Figure 4. Comparative analysis of convergence rate.

Table 1. PARAMETER SETTINGS.

Table 1(a) THE PARAMETERS USED IN BFEA

Parameter	Description	Value
PopulationSize	the population size	100
TopSize	The number of elite individuals	10
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
R_m	The range of mutation	8
P_m	The mutation ratio	0.10
Gen	Maximum number of iterations	100

Table 2(b) THE PARAMETERS USED IN CQPSO

Parameter	Description	Value
PopulationSize	The population size	100
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
C_1	Acceleration constant	2.0
C_2	Acceleration constant	2.0
V_1	Maximum velocity	+1.0
V_2	Minimum velocity	- 1.0
w	Inertia weight factor	0.5~1
Gen	Maximum number of iterations	100

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

Table 3(c) THE PARAMETERS USED IN CS

Parameter	Description	Value
Number of nests	The population size	100
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
S	Step size	1
P_m	Mutation probability	0.20
SF	Scale factor	1.5
Gen	Maximum number of iterations	100

Table 4(d) THE PARAMETERS USED IN CHBMA

Parameter	Description	Value
Q	The number of queens	2
NumberOfDrones	The number of drones	100
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
a	The decreasing factor	0.98
N_{sp}	The capacity of spermatheca	10
S(0)	The initial speed of mating flight	1.0
P_c	The breeding ratio	0.5
Gen	Maximum number of iterations	100

Table 5(e) THE PARAMETERS USED IN EPO

Parameter	Description	Value
PopulationSize	The search agents	100
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
M	The movement parameter	2
f	The control parameter	[2, 3]
L1	The control parameter	[1.5, 2]
Gen	Maximum number of iterations	100

Table 6(f) THE PARAMETERS USED IN FA

Parameter	Description	Value
PopulationSize	The population size	100
M	The number of threshold value	5,7,9,12
L	The grayscale of image	0~255
α	Randomization parameter	0.1
β	Attractiveness	0.6
ρ	Absorption coefficient	0.1
Gen	Maximum number of iterations	100

D. Breeding Operator

A new individual $\boldsymbol{X}_i$ of the current population is generated from the j^{th} elite individual $\boldsymbol{X}_j$ based on the Equation (7), which is the breeding operator of the BFEA algorithm.

$$\boldsymbol{X}_i = \boldsymbol{X}_j + uint8(randi(10,1,C)-5) \tag{7}$$

The *uint8*($\boldsymbol{Y}$) function converts the elements of the array $\boldsymbol{Y}$ into unsigned 8-bit integers. The *randi*(10,1, C) function generates *C* integer values from the uniform distribution on the set 1:10.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

E. Mutation Operator

In this stage, some new individuals will be produced by mutation operator. In each generation, $PopulationSize*P_m$ individuals are selected to mutate. The value of $\boldsymbol{X}_{id}$ (the d^{th} dimension of any individual $\boldsymbol{X}_i$) can be mutated according to formula (8). Through lots of experiments, we found that Equation (8) can help preserve population diversity. The scope of mutation R_m is set to 8. The signs (plus sign and minus sign) are equally likely to be positive or negative. If only plus sign or minus sign is used in Equation (8), the objective function value of $\boldsymbol{X}_{id}$ will always increase or decrease significantly.

$$\boldsymbol{X}_{id} = \boldsymbol{X}_{id} \pm R_m \times rand(1) \quad (8)$$

The rand(1) function randomly generates a decimal number greater than zero and less than one according to a uniform distribution. The appropriate value of R_m can speed up the algorithm convergence speed.

F. Removing the similar threshold values

In our experiment, when the absolute value of the difference between the two thresholds in the same image is less than 6, we will replace the smaller one with a larger one. This has little effect on the image quality after segmentation, but the segmentation efficiency of the proposed algorithm can be further improved. Therefore, the number of thresholds obtained after the segmentation of an image may be smaller than the number of thresholds initially set in the proposed algorithm.

G. The Parameters used in BFEA

Table 1(a) lists the parameter values in the BFEA algorithm in this experiment, with the help of a large number of tests.

IV. EXPERIMENTAL STUDIES

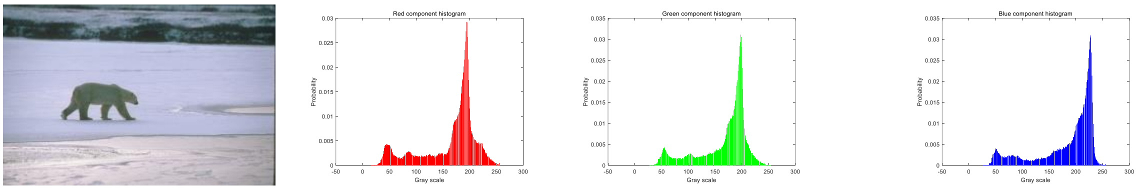
One hundred images from the Berkeley image segmentation dataset (www.eecs.berkeley.edu) have been tested to evaluate the effectiveness of the BFEA algorithm. In the experiments in this paper, we conducted a detailed study of eight color images with different characteristics to show the performance differences of different algorithms. For color image segmentation, the red, green, and blue components need to be searched for the optimal threshold separately.

Algorithm experiment platform: WINDOW 7 system; 2.20 GHz CPU; 4 GB RAM; MATLAB R2016a (win64).

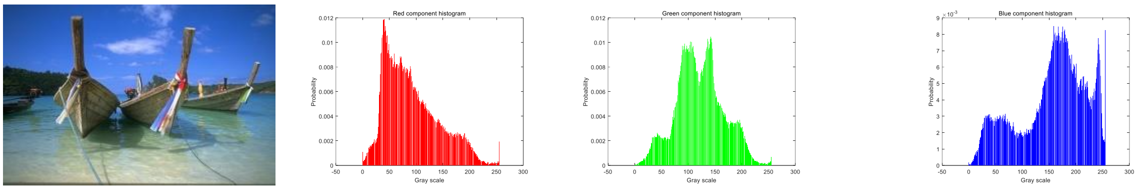
We compared the results of BFEA with some other representative algorithms so far to evaluate the performance of the proposed algorithm: quantum behavioral PSO with a cooperative method (CQPSO) [4]; Cuckoo Search (CS) method [16]; Honey Bee Mating Algorithm with a cooperative method (CHBMA) [1]; Emperor Penguin Optimizer (EPO) [17]; Firefly Algorithm (FA) [18,19].

BFEA parameters and other different algorithm parameters are shown in Table 1(a) ~ (f). The test image and corresponding histogram are presented in Figure 5. In order to evaluate the performance of the algorithms fairly, we use the same stopping criteria.

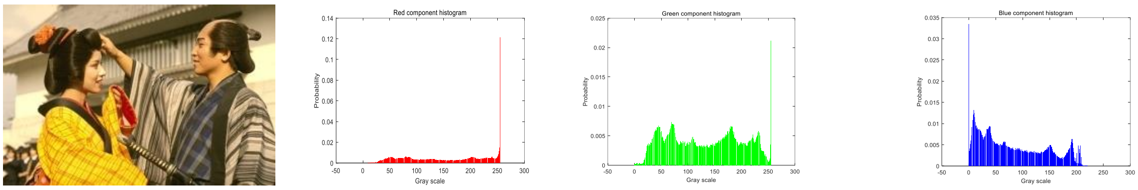
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1



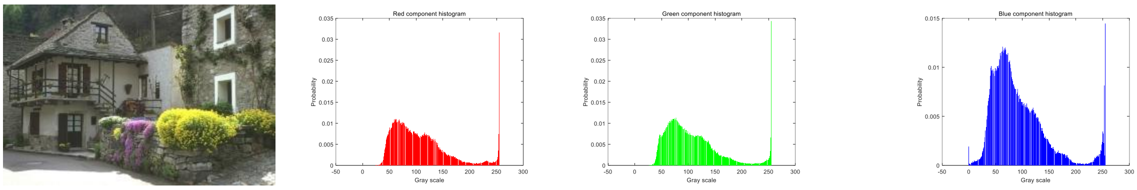
Test01



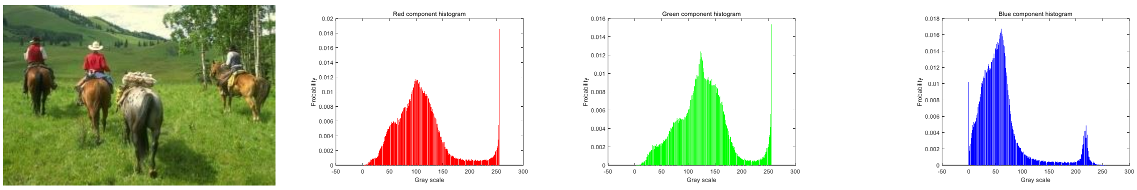
Test02



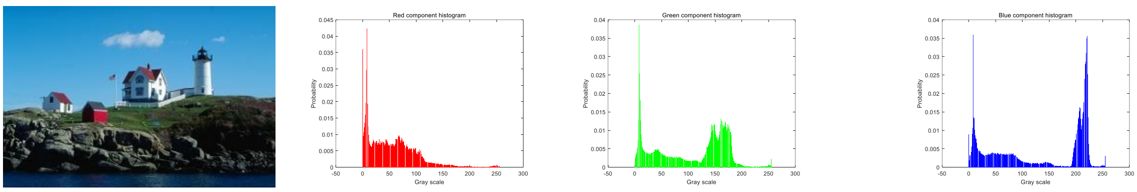
Test03



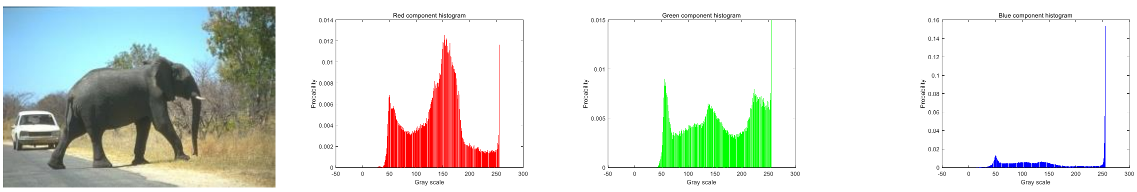
Test04



Test05

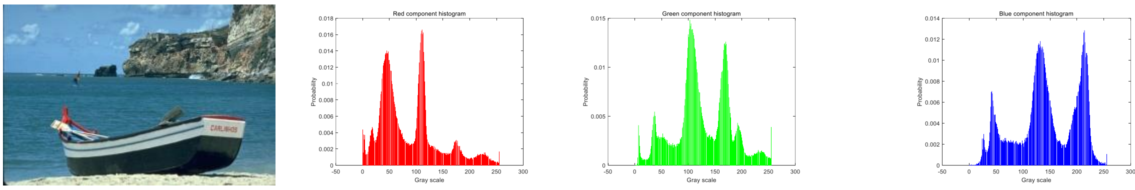


Test06



Test07

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1



Test08

Figure 5. Test images and corresponding histograms.

A. Comparison of results after segmentation

Figure 6 shows the segmented images when the threshold number is five. These results demonstrates that the BFEA algorithm has more favorable and reliable segmentation results for most of the test images from the visual effects of human eyes. From the segmented images with M = 7, 9, and 12, BFEA algorithm can also obtain better results; these segmentation results are not given because of the space limitations of the paper.

(The segmented images come respectively from these algorithms (1)BFEA (2)CQPSO (3)CS (4)CHBMA (5)EPO (6)FA from left to right.)

Figure 6. The segmented images with the threshold number is five.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

B. Comparison of Fitness, PSNR, SSIM, Computation Time

The average fitness value of the three components of an image is calculated as the Fitness Value of it.

PSNR is defined as shown in Equation (9) and is used to evaluate the image quality after multi-threshold image segmentation.

$$PSNR = 20 \times \log_{10}(255 / RMSE) \quad (9)$$

In the above formula, RMSE is defined in Equation (10) and denotes the mean-squared error.

$$RMSE = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(I(i,j) - I^{'}(i,j))^2 \Big/ (m \times n)} \quad (10)$$

In the above formula, I represents the original grayscale image of m×n and I' represents the corresponding grayscale image obtained after segmentation. The larger the *PSNR* value, the better the quality of the segmented image. For color images, we need to calculate the *PSNR* value of each component and then average them.

SSIM [20] can be used to measure the structural similarity between any two images. In general, the smaller the SSIM value, the lower the similarity between the two images. The SSIM value is calculated by the function ssim() provided by MATLAB R2016a.

For easy comparison, we take the average of the calculation time of the three components of the color image as the Computation Time.

We used six algorithms to calculate the original image and its corresponding segmented image under 5, 7, 9, 12 levels thresholding. The data in Table 2 gives the Fitness, PSNR, SSIM, and computation time. It can be seen from Table 2 that in most cases, BFEA was faster than other algorithms; its calculation time was less affected by the threshold number and can meet the needs of real-time applications. Generally, the Fitness, PSNR, or SSIM values obtained from BFEA were more competitive， which were optimal or close to optimal in most cases. This phenomenon is more apparent when the number of thresholds is large. At the same time, the results of several other algorithms are also acceptable. Besides, Table 3 contains the thresholds obtained by segmenting eight images using six algorithms.

Table 2. The comparison of Fitness, PSNR, SSIM, computation time.

Table 2 (a) *M*=5 and 7

	Method	*M*	*Fitness*	*PSNR*	*SSIM*	time(s)	*M*	*Fitness*	*PSNR*	*SSIM*	time(s)
Test01	BFEA	5	20.9267	26.7853	0.8268	**3.1618**	7	25.7270	**30.3224**	0.6792	**3.1709**
	CQPSO	5	20.8939	26.6434	0.8247	3.2379	7	25.7708	29.3955	0.7667	3.2541
	CS	5	20.2319	24.5619	0.6576	3.2136	7	25.1131	25.5606	0.5059	3.4634
	CHBMA	5	**20.9333**	26.2968	**0.8313**	3.2550	7	24.8890	28.6673	0.7088	3.2735
	EPO	5	20.8799	26.6066	0.8172	3.1850	7	**25.8342**	29.6083	**0.7921**	3.1912
	FA	5	20.4500	**28.0139**	0.7486	3.2389	7	25.7341	29.9271	0.7686	3.3106
Test02	BFEA	5	**22.1637**	25.2962	0.8480	**3.1633**	7	27.3480	**29.3357**	**0.9021**	**3.1660**
	CQPSO	5	21.7908	24.5397	0.8253	3.2839	7	26.7588	27.1744	0.7848	3.6679
	CS	5	21.7092	**25.5124**	0.8334	3.2551	7	26.6106	26.1001	0.8307	3.4728
	CHBMA	5	21.8825	24.8694	0.7710	3.1911	7	26.6046	24.4441	0.7743	3.2140
	EPO	5	22.1516	25.3882	0.8365	3.2070	7	27.2934	31.8099	0.8989	3.1904
	FA	5	22.1546	25.1303	**0.8517**	3.3578	7	**27.3827**	28.4186	0.8924	3.3967
Test03	BFEA	5	21.7175	**24.8158**	**0.7582**	3.1642	7	**26.7791**	26.4899	0.8355	**3.1580**
	CQPSO	5	21.7333	24.2337	0.7234	3.2304	7	25.9808	27.1737	0.8189	3.5639
	CS	5	21.6106	21.4347	0.6147	3.3529	7	26.0899	24.8238	0.6986	3.3153
	CHBMA	5	21.7297	24.1986	0.7480	3.1926	7	26.0759	27.1517	**0.8708**	3.2120

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

	EPO	5	21.7382	23.6552	0.7032	**3.1534**	7	26.6999	**27.8472**	0.8505	3.1898
	FA	5	**21.7453**	23.5793	0.6929	3.2811	7	26.7120	27.4226	0.8270	3.2638
Test04	BFEA	5	21.3330	24.0072	**0.7307**	**3.1658**	7	26.3463	**31.7254**	**0.8370**	**3.1500**
	CQPSO	5	21.1420	25.1407	0.6715	3.2757	7	25.7099	24.5338	0.6675	3.3238
	CS	5	21.0422	24.2705	0.7042	3.2711	7	25.8858	27.0252	0.6515	3.3132
	CHBMA	5	20.9282	**25.1432**	0.5706	3.1987	7	25.6180	29.8197	0.8136	3.2126
	EPO	5	**21.3419**	23.3134	0.7001	3.1869	7	26.3764	28.6967	0.7238	3.2357
	FA	5	21.3183	25.0924	0.6577	3.2637	7	**26.4656**	28.4723	0.8228	3.3390
Test05	BFEA	5	**21.7213**	**26.0878**	0.8348	**3.1502**	7	26.7679	**32.0380**	**0.9383**	3.1856
	CQPSO	5	21.1459	24.2389	0.7378	3.3649	7	26.2858	26.5530	0.8613	3.7118
	CS	5	21.5023	24.0152	0.7377	3.3981	7	26.4100	28.6574	0.8794	3.5769
	CHBMA	5	21.4163	23.2423	0.8167	3.1933	7	26.3407	22.3359	0.7846	3.2570
	EPO	5	21.7004	24.1121	**0.8611**	3.1848	7	**26.8064**	28.8287	0.8432	**3.1822**
	FA	5	21.7000	23.4830	0.8557	3.2547	7	26.7978	29.5970	0.9295	3.7072
Test06	BFEA	5	**21.7635**	26.6474	0.8109	**3.1154**	7	**26.9073**	29.4195	**0.8748**	**3.1724**
	CQPSO	5	21.074	22.0849	0.8046	3.2668	7	26.4915	27.6818	0.8613	3.6752
	CS	5	21.2813	23.5479	0.8087	3.3579	7	25.6051	26.4077	0.8334	3.7809
	CHBMA	5	21.2815	**27.3677**	0.8077	3.1948	7	25.8956	24.7010	0.7609	3.2260
	EPO	5	21.7633	25.6461	0.8025	3.1871	7	26.7965	28.8550	0.8588	3.2661
	FA	5	21.7502	26.8280	**0.8187**	3.4471	7	26.7869	**30.2522**	0.8715	3.7172
Test07	BFEA	5	**21.0531**	**27.7074**	0.6792	**3.1313**	7	25.4661	**31.3393**	**0.8326**	**3.1563**
	CQPSO	5	20.6462	26.2506	0.6715	3.2908	7	25.9269	28.3979	0.7070	3.3263
	CS	5	20.3756	26.6501	**0.7446**	3.2240	7	24.9012	27.0113	0.8129	3.4585
	CHBMA	5	20.5992	27.1824	0.7239	3.1640	7	25.0035	27.9884	0.7203	3.1997
	EPO	5	21.0439	27.3554	0.6565	3.2411	7	**25.9281**	29.4240	0.7847	3.1715
	FA	5	21.0263	27.6009	0.6696	3.3349	7	25.8856	27.2460	0.6823	3.4842
Test08	BFEA	5	21.6896	23.6057	**0.8695**	**3.1428**	7	**26.9441**	**29.4836**	**0.9282**	**3.1732**
	CQPSO	5	21.3574	21.1083	0.7382	3.3433	7	26.0737	26.1329	0.8791	3.4605
	CS	5	21.4051	22.4917	0.8124	3.3342	7	26.0094	25.1890	0.7788	3.5990
	CHBMA	5	21.3791	**26.3505**	0.8242	3.1703	7	26.2456	26.4758	0.8652	3.2179
	EPO	5	**21.7135**	22.1515	0.8423	3.1927	7	26.9089	29.2625	0.9270	3.2023
	FA	5	21.6856	24.0007	0.8632	3.6090	7	26.8751	26.3868	0.8769	4.0572

Table 2 (b) *M*=9 and 12

	Method	*M*	*Fitness*	*PSNR*	*SSIM*	time(s)	*M*	*Fitness*	*PSNR*	*SSIM*	time(s)
Test01	BFEA	9	**30.2386**	**32.3422**	**0.8622**	**3.1762**	12	**36.1143**	31.1440	**0.8911**	**3.1768**
	CQPSO	9	29.0449	25.7962	0.7082	3.5848	12	35.9684	27.5292	0.7153	4.2449
	CS	9	29.1578	26.7619	0.4489	3.5813	12	34.4821	31.4188	0.7247	3.8909
	CHBMA	9	29.0744	28.4829	0.7177	3.2777	12	34.1074	**31.7630**	0.7845	3.3482
	EPO	9	30.2181	28.7226	0.7147	3.2103	12	36.0918	30.6706	0.7790	3.2194
	FA	9	30.0031	31.8823	0.8383	3.4448	12	36.0149	30.9964	0.7760	3.4472
Test02	BFEA	9	**32.0051**	32.0989	**0.9116**	**3.1654**	12	**38.0707**	**34.2178**	**0.9206**	**3.1759**
	CQPSO	9	31.1288	25.5111	0.8904	3.9724	12	36.3870	26.7466	0.8278	4.5133
	CS	9	31.0486	26.9562	0.7658	3.8477	12	38.0117	33.1314	0.9158	4.4970
	CHBMA	9	31.2028	26.4347	0.7949	3.2427	12	36.7947	29.6582	0.8203	3.3311
	EPO	9	31.9680	32.0426	0.8921	3.1880	12	38.0429	31.3207	0.9051	3.2269
	FA	9	31.8981	**33.1542**	0.9034	3.4490	12	38.0496	31.0363	0.8986	3.5294
Test0	BFEA	9	31.1603	**29.9751**	**0.9332**	**3.1668**	12	**37.2442**	**31.4737**	**0.9410**	3.1934
	CQPSO	9	30.1642	24.6714	0.7879	3.6389	12	35.5970	27.6417	0.8842	4.0671

	CS	9	29.7995	23.8646	0.7330	3.5667	12	35.2974	27.5683	0.8681	4.1493
	CHBMA	9	29.8704	28.0127	0.8826	3.1791	12	35.3297	24.3518	0.6104	3.2658
	EPO	9	28.5692	22.9372	0.7050	3.2061	12	37.1982	30.7508	0.9055	**3.1866**
	FA	9	**31.2888**	27.6472	0.8290	3.4405	12	37.0371	29.4894	0.8476	3.5269
Test04	BFEA	9	**31.0283**	32.6399	**0.8758**	**3.1618**	12	**36.8963**	31.4685	0.8186	**3.1813**
	CQPSO	9	29.9318	28.6811	0.7039	3.6037	12	35.1844	28.8931	**0.8546**	4.3826
	CS	9	29.3458	28.5956	0.7097	3.7419	12	34.8870	26.6505	0.7100	4.5737
	CHBMA	9	30.0698	26.7191	0.7189	3.2013	12	35.3333	30.3050	0.8308	3.2031
	EPO	9	30.8441	32.6809	0.8502	3.1847	12	36.7139	29.6120	0.8277	3.1913
	FA	9	30.8799	**33.0989**	0.8619	3.3924	12	36.8195	**31.6197**	0.8399	3.5194
Test05	BFEA	9	**31.5614**	31.8859	**0.9511**	**3.1885**	12	**37.7524**	33.5213	**0.9587**	**3.1855**
	CQPSO	9	30.4021	26.0289	0.8604	4.2261	12	36.3925	28.2916	0.8423	4.6050
	CS	9	30.4163	29.1901	0.9200	3.6611	12	36.0845	27.4874	0.8801	4.5651
	CHBMA	9	30.4864	25.9480	0.8139	3.2489	12	36.0159	29.2902	0.9165	3.2211
	EPO	9	31.4196	31.4635	0.9390	3.2124	12	37.3884	31.5442	0.9440	3.2290
	FA	9	31.5538	**32.5884**	0.9483	4.2941	12	37.7421	**34.0384**	0.9574	4.7562
Test06	BFEA	9	**31.4461**	31.4425	**0.9011**	**3.1567**	12	**37.5630**	**33.9354**	**0.9277**	**3.1892**
	CQPSO	9	30.0712	29.5179	0.8696	3.7682	12	35.7594	27.7475	0.8793	4.4167
	CS	9	29.8671	25.5442	0.8254	3.8396	12	35.8541	26.9032	0.8748	4.4816
	CHBMA	9	30.3047	28.5039	0.8645	3.2406	12	35.6432	27.7639	0.8578	3.3503
	EPO	9	31.3930	**31.9947**	0.8988	3.1653	12	37.3791	32.4631	0.9133	3.2185
	FA	9	31.2786	30.3369	0.8883	4.4107	12	37.4463	32.9797	0.9162	4.7002
Test07	BFEA	9	**30.4208**	**31.2342**	**0.7895**	**3.1716**	12	36.0679	29.4435	**0.8973**	**3.1726**
	CQPSO	9	29.1524	27.7036	0.7258	3.3593	12	33.9000	28.1894	0.7838	3.8821
	CS	9	28.8245	29.2613	0.7633	3.5781	12	34.4768	26.8675	0.7466	3.6591
	CHBMA	9	28.9345	27.2605	0.7056	3.2019	12	34.0476	30.7233	0.8423	3.1995
	EPO	9	30.2347	26.1277	0.7245	3.1864	12	36.2200	**33.5316**	0.8929	3.1834
	FA	9	30.2894	29.7235	0.7322	3.7558	12	**36.3085**	29.2905	0.8045	4.1808
Test08	BFEA	9	**31.5319**	**33.2659**	**0.9531**	**3.1798**	12	**37.6560**	**31.8913**	**0.9331**	**3.1931**
	CQPSO	9	30.5670	24.5595	0.8467	3.7160	12	35.9282	29.1305	0.8635	4.4417
	CS	9	30.5044	25.8077	0.7641	4.0679	12	35.7255	30.8571	0.9323	4.6585
	CHBMA	9	30.3264	28.7129	0.8986	3.1968	12	36.2003	30.8362	0.9256	3.2933
	EPO	9	31.5194	32.6567	0.9495	3.1829	12	37.5805	30.7359	0.9214	3.2027
	FA	9	31.4778	32.3002	0.9332	4.4963	12	37.6099	30.2860	0.8903	4.7802

Table 3. The optimal threshold value of each algorithm.

Table 3 (a)

	M	BFEA	CQPSO	BA
Test01	5	67 102 136 168 207 75 111 141 176 213 73 111 146 177 207	71 108 142 168 211 79 109 141 173 219 76 111 146 176 205	78 111 169 189 221 42 80 110 186 208 37 85 110 125 187
	7	36 73 102 134 166 188 217 45 74 101 133 162 187 219 37 75 107 141 172 202 239	58 92 122 149 174 204 238 43 76 108 139 165 191 222 37 78 105 135 159 186 210	62 96 115 171 196 219 234 35 48 71 109 150 163 212 38 74 107 132 161 206 240
	9	36 57 84 117 149 167 186 209 237 44 73 93 112 132 153 182 210 231	23 39 55 81 94 124 152 220 239 47 73 91 117 133 152 171 207 246	33 79 98 135 160 173 186 225 234 40 59 83 101 128 135 154 185 229

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

		37 64 87 106 125 151 177 203 221	33 44 67 76 87 131 144 182 210	37 76 95 157 171 187 207 225 242
	12	37 57 73 91 114 135 155 169 187 203 219 239 44 71 93 108 123 145 156 171 194 209 224 241 38 60 74 91 110 125 147 170 194 210 225 245	29 50 65 84 102 122 141 167 188 206 227 245 43 66 86 107 125 137 160 175 190 211 230 246 61 81 96 111 132 157 175 194 206 221 236 247	33 41 54 69 99 110 136 156 182 190 199 40 61 83 96 115 139 149 169 193 208 225 38 59 72 86 97 113 147 167 176 203 238
Test02	5	31 81 128 170 222 34 81 124 170 224 51 94 137 176 217	39 82 119 160 211 16 85 125 168 227 59 107 130 156 204	32 86 168 183 223 23 82 129 169 218 58 83 140 183 202
	7	32 63 96 132 163 198 226 32 69 100 127 158 194 224 26 58 92 125 157 189 224	26 78 91 113 135 165 234 34 70 99 121 146 173 221 35 66 117 160 177 195 229	19 52 104 157 176 202 234 36 62 91 121 134 159 217 45 74 108 127 150 199 211
	9	31 64 90 120 145 168 192 215 233 27 54 78 99 127 149 173 201 233 23 52 80 110 135 163 188 211 232	25 58 73 102 110 125 155 180 221 23 42 73 97 129 136 157 183 218 14 45 64 82 131 140 158 177 209	10 33 71 87 103 128 159 206 239 27 52 71 87 130 170 195 209 242 17 54 66 92 145 169 185 208 241
	12	16 34 59 85 107 125 140 160 175 192 215 234 29 53 71 87 107 121 140 174 193 212 229 243 20 42 62 82 102 122 142 163 181 200 222 241	27 49 65 77 116 137 147 169 192 215 237 18 25 40 57 67 89 107 152 165 182 228 248 8 30 44 59 77 89 117 125 163 196 224 247	29 49 67 84 104 128 150 171 188 203 222 237 22 49 68 85 106 138 158 177 198 218 231 243 23 51 74 91 109 127 147 161 176 189 209 236
Test03	5	52 97 142 185 219 43 84 127 167 211 43 78 113 144 178	62 102 143 183 221 21 82 125 167 211 43 78 110 143 176	62 101 142 175 205 20 65 116 157 189 28 50 84 131 178
	7	30 65 98 131 164 192 224 22 53 87 122 154 195 225 32 64 89 114 137 160 183	44 63 88 124 178 204 243 32 71 113 126 160 180 214 20 70 91 119 133 175 195	36 63 76 126 160 184 219 64 111 133 151 182 215 229 21 47 89 100 122 163 183
	9	29 52 74 100 132 162 188 213 233 24 52 77 98 124 149 177 200 228 31 53 75 94 113 136 169 198 216	28 44 71 85 116 125 155 193 230 31 46 88 129 152 173 195 216 223 20 31 46 57 93 118 154 164 210	25 45 58 69 96 144 169 203 246 21 60 72 98 144 155 162 187 208 29 36 66 81 105 112 124 144 186
	12	28 44 64 81 97 111 127 147 167 186 208 230 21 45 68 87 106 125 146 159 179 199 222 237 20 44 62 75 88 101 117 139 165	21 34 45 74 100 117 132 150 177 198 213 27 46 58 66 82 106 123 150 157 170 206 217 11 22 53 73 96 104 112 141 156 177 200 226	13 23 30 55 90 109 121 143 171 192 210 247 21 71 89 114 125 145 168 175 193 225 240 23 53 79 108 132 143 160 173

		180 201 216		185 191 205
Test04	5	33 83 120 155 192 38 77 116 152 187 33 89 143 196 232	28 71 115 158 203 67 91 128 154 201 54 94 158 191 231	49 89 125 159 212 34 88 114 138 175 29 96 132 186 237
	7	33 71 106 139 172 200 226 40 71 103 133 167 201 226 33 70 110 148 175 198 232	32 50 65 108 124 184 202 40 64 77 120 159 195 220 43 79 99 160 185 212 233	28 69 103 146 188 210 235 35 74 104 141 167 220 237 27 52 81 102 136 167 197
	9	34 59 82 103 126 153 182 204 226 38 63 82 107 131 160 186 209 233 32 56 81 107 131 158 182 204 232	29 66 90 122 174 191 205 216 231 33 62 82 121 132 139 167 184 232 24 38 74 106 118 128 166 200 233	38 53 68 105 116 138 186 199 238 24 40 100 133 148 161 198 208 235 34 83 110 121 142 148 180 209 240
	12	29 41 59 79 97 117 136 155 184 204 226 241 37 60 78 95 110 125 141 156 174 198 217 236 28 50 72 102 123 142 158 180 197 218 230 246	39 58 90 97 121 145 159 193 208 215 228 38 45 81 94 111 132 161 175 187 200 236 37 58 77 98 114 127 155 174 186 222 239	36 44 71 97 112 133 152 166 186 212 240 37 80 101 116 128 151 162 195 219 237 40 56 77 83 108 136 158 167 183 206 230

Table 3 (b)

	M	CHBMA	EPO	FA
Test01	5	66 103 140 168 206 78 113 144 172 211 71 103 138 172 205	74 107 138 168 215 76 112 149 176 217 73 109 143 174 202	81 109 131 178 214 83 103 155 173 221 68 111 151 202 216
	7	38 49 91 111 161 184 216 57 71 106 134 171 209 246 38 48 93 130 155 206 221	28 71 102 135 168 204 233 43 78 107 135 164 188 221 37 72 105 138 161 187 212	36 71 108 142 175 208 235 44 72 98 126 155 184 219 37 69 102 135 169 200 219
	9	30 57 73 102 133 155 183 199 221 75 81 99 126 143 157 201 220 237 61 95 102 113 136 144 173 195 218	35 65 90 122 150 173 201 220 237 44 68 88 107 129 149 170 192 225 37 64 85 110 129 151 175 200 221	36 59 85 116 143 166 186 211 238 48 78 105 124 143 166 184 214 237 66 87 109 128 147 166 187 206 224
	12	22 31 43 65 98 109 128 138 162 212 233 32 51 78 106 115 123 132 146 167 209 230 37 53 71 81 94 126 141 148 169 205 222	35 54 74 91 109 123 146 162 184 203 222 236 44 64 75 95 116 133 147 163 176 193 217 233 38 69 98 124 141 154 172 184 198 211 224 242	34 53 64 84 104 131 145 160 174 187 206 234 42 53 75 90 110 133 157 176 196 216 230 245 38 64 82 95 114 134 157 170 195 209 222 242
Test02	5	30 105 129 162 222 49 102 147 172 218 50 86 118 146 201	31 76 125 173 222 32 81 133 174 224 56 101 137 175 215	31 83 132 180 222 31 75 124 170 226 54 95 133 173 213
	7	9 51 78 102 130 181 218 22 57 100 130 149 211 238 37 69 85 102 138 163 206	31 69 101 135 165 196 229 30 76 111 143 172 199 230 45 78 110 138 167 195 224	30 67 104 136 172 202 227 27 58 87 123 158 192 224 24 57 91 122 154 190 223
	9	15 46 56 93 129 150 173 199 221	30 52 73 98 123 147 178 203 228	31 59 84 107 135 163 191 213 236

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2024 doi:10.20944/preprints202406.1791.v1

		22 36 74 105 114 127 165 182 224 21 58 87 114 153 185 200 227 237	25 58 89 114 137 164 197 216 235 26 56 80 108 133 156 182 211 233	25 56 80 111 145 179 202 218 235 29 59 86 111 133 159 188 212 235
	12	11 26 58 71 96 120 139 153 178 193 230 27 41 65 98 119 131 161 183 203 219 236 24 52 59 77 98 129 148 155 162 172 198 226	31 60 81 97 117 136 149 167 181 199 219 242 25 43 63 78 93 116 137 158 180 200 221 235 22 38 60 80 108 138 160 179 195 211 224 238	31 54 75 93 112 134 149 167 188 207 226 238 24 46 63 82 96 114 130 150 176 197 219 239 22 61 88 106 124 144 162 176 193 209 225 240
Test03	5	52 96 140 183 220 21 77 123 169 211 47 78 110 143 178	63 104 146 183 219 22 68 116 162 209 43 77 110 142 178	65 105 144 183 220 21 67 116 163 210 43 77 110 141 176
	7	28 66 96 153 182 201 229 34 71 83 111 138 185 228 38 53 71 111 164 179 219	29 64 102 135 164 197 226 21 57 100 142 169 199 229 37 67 95 120 147 165 185	29 68 102 141 175 205 230 22 62 95 130 163 197 224 30 62 90 116 140 167 188
	9	41 57 70 86 95 129 156 197 226 24 62 83 104 125 175 199 225 235 11 54 72 105 122 183 198 214 222	57 72 98 118 136 152 171 186 213 18 39 56 77 111 129 151 171 203 24 52 79 110 134 151 180 198 218	24 50 72 101 128 156 183 208 232 23 50 80 108 139 167 191 212 234 24 50 71 89 110 129 149 169 186
	12	26 46 69 93 135 150 168 179 207 214 236 20 43 51 85 92 99 111 146 156 166 200 224 6 22 31 43 67 74 92 121 133 169 208 216	24 45 61 79 99 129 152 175 189 203 222 238 20 39 58 79 102 116 132 157 174 194 212 234 18 34 52 71 92 110 128 144 162 180 188 201	24 43 64 85 104 131 153 172 191 207 226 241 21 41 60 82 98 118 132 164 185 203 220 241 28 41 50 67 85 102 115 131 149 165 184 199
Test04	5	64 95 144 176 200 57 102 147 165 180 34 72 102 168 233	33 73 110 148 187 37 78 115 152 188 34 98 150 195 234	37 75 115 155 193 66 98 131 160 191 34 93 149 195 232
	7	54 78 91 122 144 168 226 48 64 101 129 173 191 216 48 77 106 135 146 193 238	36 62 91 122 153 186 219 36 82 121 151 181 205 230 30 60 95 126 164 205 232	39 70 96 125 157 192 221 37 64 95 128 157 187 219 31 58 90 123 161 197 233
	9	27 56 83 99 122 139 169 185 198 30 70 78 89 106 128 172 206 225 28 42 57 104 154 171 191 221 237	33 59 87 115 142 162 186 211 229 39 72 98 124 149 168 192 212 239 34 66 87 112 137 166 200 221 239	36 61 88 110 133 155 180 205 227 37 61 85 118 149 173 196 218 237 30 62 92 122 150 182 208 227 241
	12	38 64 70 81 91 115 138 167 194 215 226 39 50 63 75 103 123 155 179 208 216 227 32 60 86 102 109 123 159 176	35 61 78 96 112 130 143 161 180 202 218 234 40 72 87 102 117 133 146 160 178 194 217 238	29 41 62 86 104 116 140 161 180 202 225 237 33 51 67 85 103 119 134 151 168 191 213 235

 doi:10.20944/preprints202406.1791.v1

		184 207 236	33 50 72 89 111 140 162 185 198 220 232 245	30 52 78 97 129 156 172 184 197 212 233 245

Table 3 (C)

	M	BFEA	CQPSO	BA
Test05	5	37 79 121 166 212 59 102 144 192 228 44 90 131 169 208	27 89 124 163 182 62 100 172 197 222 21 81 111 142 168	26 72 120 157 195 46 112 150 193 223 31 70 125 160 199
	7	37 74 114 147 176 207 236 45 73 108 142 174 206 234 50 90 124 151 181 208 231	29 52 97 109 164 199 237 34 55 95 156 182 214 236 32 54 79 111 141 183 216	31 86 117 153 175 205 226 19 52 88 134 156 194 212 27 71 97 132 150 163 209
	9	33 56 80 108 135 162 187 210 235 20 55 80 103 124 149 176 202 234 35 62 86 112 146 169 190 209 231	15 28 71 97 148 177 195 216 231 27 60 88 101 114 135 144 171 227 62 81 106 136 147 183 199 211 234	20 54 62 102 119 131 155 203 218 19 50 72 104 147 167 194 230 247 29 58 86 101 124 136 177 191 221
	12	31 50 73 91 110 127 145 164 184 204 220 237 20 45 68 89 108 123 139 154 170 192 216 239 22 49 66 87 106 128 162 179 195 211 231 243	33 79 85 92 111 135 145 165 179 200 210 227 31 47 63 77 100 110 157 172 181 202 221 245 12 30 37 65 87 114 131 149 167 194 216 233	17 25 48 68 102 120 153 168 192 207 241 255 12 36 43 59 87 98 127 135 142 167 195 218 35 56 72 95 105 114 128 145 171 183 209 217
Test06	5	42 85 123 178 219 60 98 134 187 218 53 106 157 195 232	18 83 125 173 228 40 83 135 146 211 39 73 103 166 189	62 95 123 171 199 33 64 130 162 203 42 80 107 142 197
	7	36 64 95 123 153 181 219 41 71 105 134 164 187 213 41 65 98 131 161 194 232	44 77 117 173 186 209 241 21 54 96 120 142 183 219 35 64 94 117 150 196 228	29 70 102 127 173 225 238 60 80 97 114 129 182 219 42 54 73 87 167 200 238
	9	34 67 90 118 145 173 195 218 238 30 64 90 112 131 154 191 213 236 36 63 94 113 136 160 178 193 231	54 80 95 111 134 157 174 220 249 32 81 93 113 154 179 185 212 241 31 51 82 100 141 166 185 204 240	14 39 48 88 121 159 182 194 209 10 30 89 116 141 150 193 210 225 26 43 56 62 102 114 134 188 230
	12	22 45 66 87 104 121 141 160 181 200 224 243 31 52 68 86 106 126 141 156 172 188 208 235 21 43 63 91 108 124 139 152 170 194 215 231	25 40 47 82 111 121 137 147 173 196 211 221 27 46 88 96 129 138 159 193 212 226 238 255 17 59 73 100 117 142 166 187 213 227 243 255	10 31 74 102 112 122 142 159 191 218 226 239 21 60 88 98 130 165 177 186 199 207 223 236 34 41 59 80 95 126 153 179 197 218 229 255
Test07	5	81 116 149 187 225 82 117 152 188 220 40 83 126 169 208	97 118 182 206 234 68 102 143 183 212 29 101 150 174 199	64 83 140 166 221 84 114 137 208 221 62 95 111 156 191
	7	56 83 105 127 168 194 222 50 70 103 141 159 214 230	40 77 108 146 183 211 236 77 104 132 160 190 214 240	48 96 135 173 191 209 224 48 76 124 169 180 194 238

		39 89 124 147 169 207 236	38 67 99 128 156 185 216	47 77 106 169 186 216 245
	9	40 80 101 122 141 165 186 208 234 52 77 100 120 140 161 183 210 238 31 47 73 95 126 154 180 202 229	37 68 77 111 144 177 189 202 220 88 101 109 117 125 146 170 192 241 27 36 80 113 133 150 165 192 219	44 66 81 127 150 165 173 190 210 59 78 100 131 155 171 207 214 27 41 88 116 127 154 176 223 230
	12	42 63 75 94 109 129 150 168 190 212 230 244 51 73 89 102 112 131 155 179 206 221 236 246 39 59 77 96 114 132 152 174 193 208 232 250	34 40 55 65 99 117 140 152 202 241 248 255 42 57 71 108 118 136 152 192 213 225 236 252 18 41 63 84 111 127 135 162 181 200 223	32 71 99 114 121 131 153 169 186 211 233 255 59 70 78 114 138 146 156 184 206 218 237 245 29 47 59 89 100 114 137 166 201 226 244 255
Test08	5	34 75 125 170 212 55 92 126 160 210 22 84 118 157 201	54 121 144 188 215 47 82 128 167 209 19 41 79 122 164	46 70 103 141 194 34 99 145 184 222 19 79 111 146 179
	7	35 73 102 130 160 192 224 32 73 96 127 162 202 235 22 68 99 123 156 200 236	40 66 106 158 186 219 240 30 70 92 105 126 202 217 20 65 88 116 184 216 240	19 65 85 105 126 146 204 29 57 87 104 182 212 232 32 103 118 152 168 191 241
	9	36 67 101 125 149 170 190 215 240 33 63 91 116 136 166 184 210 234 24 55 89 114 139 166 194 213 239	37 60 69 124 139 159 199 212 234 32 50 76 98 128 166 193 202 228 16 35 68 89 103 113 130 176 203	16 22 54 90 122 150 175 196 241 16 31 55 94 113 123 182 208 232 54 80 104 130 149 167 188 199 239
	12	18 41 66 84 101 124 140 164 191 213 230 240 31 62 81 96 114 134 153 166 178 198 214 241 22 39 60 78 95 110 121 148 175 200 229 243	19 39 46 66 91 129 135 155 167 173 195 236 24 51 58 86 112 126 137 145 169 204 228 20 44 65 90 104 150 160 179 202 233 242	19 30 45 75 111 138 150 177 192 209 245 10 41 53 66 99 123 132 151 181 211 223 8 28 59 76 108 116 146 172 199 206 223 245

Table 3 (D)

	M	CHBMA	EPO	FA
Test05	5	33 102 137 177 212 38 61 98 152 187 47 79 95 141 196	40 84 130 174 216 39 76 114 155 193 41 88 130 171 208	37 80 120 166 212 45 77 111 150 192 43 96 140 174 208
	7	20 40 61 87 133 184 228 29 72 95 130 149 171 209 22 39 72 105 139 168 214	35 67 98 130 160 193 223 55 86 116 146 175 204 231 30 58 90 124 158 190 211	37 69 100 129 165 200 233 22 57 89 123 157 195 234 44 81 115 153 183 208 236
	9	18 41 102 136 151 165 185 204 238 42 56 81 99 111 139 191 209 234 28 35 54 73 115 160 183 196 229	34 58 82 101 118 147 174 209 235 20 58 84 118 140 167 190 212 238 27 58 88 117 145 168 189 211 243	31 53 79 103 130 156 178 206 229 32 60 85 112 140 167 190 216 239 32 57 84 109 140 163 189 211 238

	12	17 33 46 69 106 142 161 168 180 193 225 255 11 35 52 58 87 123 152 158 181 209 220 243 14 21 53 79 96 109 150 175 200 208 234 240	36 62 77 94 109 123 145 162 193 209 229 243 28 52 71 93 113 135 161 183 202 216 233 244 26 46 71 92 112 131 150 164 180 197 213 241	26 43 62 83 104 127 144 165 182 201 225 242 20 45 62 82 108 129 146 160 178 205 230 243 27 45 64 81 98 114 131 150 168 192 211 238
Test06	5	66 92 124 177 227 66 90 144 187 221 48 124 155 196 225	43 84 128 177 219 55 96 131 161 202 60 107 155 195 231	47 86 129 170 210 56 95 134 187 218 50 99 143 196 231
	7	67 105 120 153 181 208 242 56 104 118 134 166 181 214 25 48 64 81 119 160 195	33 61 86 120 153 184 224 52 78 107 133 161 187 218 38 77 109 131 175 194 231	48 84 117 154 182 210 237 42 76 104 133 164 202 233 33 73 112 138 169 195 231
	9	29 61 100 118 136 153 166 201 238 15 66 86 104 126 141 169 194 205 27 50 90 137 154 190 206 227 255	31 58 81 120 145 170 195 217 238 27 56 87 113 138 162 187 218 235 37 68 102 136 162 179 193 207 230	33 67 100 124 151 174 195 218 238 33 68 103 125 147 172 190 213 235 34 61 84 98 112 133 167 193 230
	12	20 46 59 72 79 97 110 116 139 194 210 226 12 19 32 57 74 88 142 167 183 204 226 243 8 28 42 72 95 102 119 139 195 210 236 245	30 50 67 86 103 120 145 167 189 207 227 244 33 54 76 99 118 135 156 174 189 209 231 240 24 38 57 75 99 121 138 158 172 195 214 233	29 46 65 86 101 119 143 161 181 208 231 243 35 60 79 97 118 135 153 169 184 203 223 240 29 48 71 87 103 126 155 167 179 192 213 231
Test07	5	40 68 121 193 219 61 99 143 196 222 91 123 151 170 207	82 120 152 186 225 86 121 158 193 224 40 84 124 170 208	82 121 153 186 225 86 121 158 196 224 40 88 129 174 208
	7	39 47 93 112 171 198 224 46 90 119 146 193 214 248 40 55 103 119 145 175 210	40 79 116 145 183 208 232 78 106 128 153 179 208 232 45 78 109 142 173 206 234	42 74 105 134 165 198 227 83 114 145 172 193 214 238 40 74 107 137 167 197 229
	9	64 87 110 129 154 163 180 209 216 47 96 118 130 150 170 181 195 206 18 51 80 94 125 145 194 220 243	39 71 104 126 145 169 195 226 242 83 110 132 151 169 187 203 221 238 42 71 99 122 143 170 193 214 234	42 70 100 123 142 160 187 215 238 52 85 108 130 147 168 193 220 242 33 45 70 96 118 141 172 206 240
	12	44 71 81 91 118 124 165 173 195 214 222 231 30 42 53 74 108 118 147 162 184 205 222 228 27 41 59 73 126 139 147 168 186 193 207 218	40 59 72 88 102 126 143 159 177 194 227 242 43 54 76 92 108 128 151 165 183 199 222 239 31 44 65 91 113 128 150 174 199 218 235 250	41 67 87 98 108 126 144 161 187 204 225 241 50 72 92 111 124 145 166 186 203 216 228 243 29 45 68 84 101 128 147 174 200 221 234 249
T	5	36 96 157 194 225	33 68 101 149 199	34 75 128 182 218

 doi:10.20944/preprints202406.1791.v1

		38 85 138 161 202 76 102 147 186 230	55 93 127 160 207 22 70 117 162 197	34 95 134 181 216 23 73 118 158 198
	7	58 103 121 143 174 205 233 24 45 89 130 155 200 246 17 48 90 109 138 184 239	34 67 101 128 162 191 223 31 65 95 126 160 212 236 38 69 96 122 163 193 233	30 58 80 102 135 171 213 33 70 98 129 158 205 232 24 63 94 123 159 193 231
	9	12 54 85 107 131 154 196 207 228 18 28 50 104 139 158 187 206 227 30 47 63 82 107 167 198 231 239	32 58 81 104 132 165 193 215 241 32 60 91 115 140 159 190 214 238 24 54 86 110 126 153 184 203 241	33 54 76 101 123 149 170 197 234 35 70 94 117 144 159 182 204 240 25 61 85 112 137 162 191 210 234
	12	16 48 76 100 141 159 166 192 207 224 235 255 20 42 72 78 93 119 139 160 182 208 215 239 8 32 57 98 110 124 148 166 176 182 204 233	23 39 55 69 88 103 131 160 186 202 218 237 36 59 75 88 100 115 133 148 163 184 206 232 14 24 48 77 104 125 144 163 184 203 228 241	20 33 50 66 85 102 129 163 180 197 221 240 16 32 57 77 97 118 138 162 186 208 223 240 23 55 76 89 111 135 161 175 191 209 230 242

C. Steadiness of algorithm (M=5,7,9,12)

The standard deviation of optimal fitness value of the red component for each image is mentioned in Table 4. Under the same circumstances, the larger the standard deviation (*std*) value, the more unstable the experimental results are. From the experimental results above, in general we found that EPO and FA are superior to the other three algorithms, so in the stability test, we only use EPO and FA to compare with the proposed algorithm. Table 4 lists the standard deviation values obtained by running the three algorithms BFEA, EPO, and FA one hundred times independently, when the threshold number M = 5, 7, 9, and 12. We can draw a conclusion from the experimental data that BFEA is more efficient and stable than EPO and FA on most images.

Table 4. The Stability with M=5, 7, 9, 12 (It's the standard deviation in the table.).

M		Test01	Test02	Test03	Test04	Test05	Test06	Test07	Test08
5	BFEA	**0.0083**	**0.0113**	**0.0463**	**0.0228**	**0.0119**	**0.0214**	**0.0323**	**0.0109**
	EPO	0.0960	0.1186	0.0983	0.1152	0.1323	0.1207	0.1350	0.1048
	FA	0.1641	0.1542	0.2156	0.1434	0.1391	0.1362	0.1139	0.1351
7	BFEA	**6.1202e-04**	**0.0143**	**0.0048**	**0.0188**	**0.0172**	**0.0415**	**0.0788**	**0.0313**
	EPO	0.1039	0.1321	0.1220	0.1089	0.1575	0.1428	0.1691	0.1215
	FA	0.2361	0.1730	0.1979	0.1560	0.1764	0.2015	0.1826	0.1567
9	BFEA	**0.0622**	**0.0475**	**0.0218**	**0.0038**	**0.0436**	**0.0822**	**0.1027**	**0.0277**
	EPO	0.1395	0.1288	0.1419	0.1253	0.1768	0.1746	0.2175	0.1175
	FA	0.1624	0.1533	0.2660	0.2018	0.2534	0.2362	0.2411	0.1698
12	BFEA	**0.0119**	**0.0658**	**0.0491**	**0.0048**	**0.0271**	**0.0723**	**0.1208**	**0.0318**
	EPO	0.1227	0.1516	0.1553	0.1461	0.1513	0.1655	0.2396	0.1437
	FA	0.1898	0.1928	0.2482	0.2067	0.2680	0.2425	0.2936	0.2168

 doi:10.20944/preprints202406.1791.v1

The algorithm proposed in this paper has achieved excellent results for two main reasons: (1) An appropriate new population in each generation is introduced to avoid premature convergence and fall into local optimum. (2) The prediction method helps search more accurately and more quickly.

D. Experiment to conquer the curse of dimensionality (M= 9, 12, 15, 18)

We also tested the computational power of the proposed method for the red component of each image in a higher dimension, where M = 9, 12, 15, 18. Table 5 lists the average calculation time (seconds) of one hundred runs using the BFEA algorithm. The experiment shows that when there are many thresholds, the efficiency of the BFEA algorithm is also excellent, which can solve the problem of high-resolution image segmentation to a certain extent.

Table 5. The mean computation time with m=9, 12, 15, 18.

M	Test01	Test02	Test03	Test04	Test05	Test06	Test07	Test08
9	3.1679	3.1612	3.1697	3.1627	3.1839	3.1532	3.1753	3.1785
12	3.1742	3.1746	3.1879	3.1795	3.1863	3.1856	3.1762	3.1762
15	3.2135	3.1960	3.1885	3.1836	3.1965	3.2053	3.1775	3.1901
18	3.3493	3.3551	3.2541	3.2209	3.2143	3.2386	3.1849	3.2623

V. CONCLUSIONS

In this paper, we described the Bayesian Forecasting Evolutionary Algorithm for color image segmentation. The optimal solution prediction technology in the algorithm can improve the efficiency of the algorithm. This algorithm is suitable for the compression technology of image or video transmission. In addition to evaluating the eight benchmark images shown above, a variety of images provided by the Berkeley segmentation data set were also tested. The experimental results showed that the proposed algorithm could usually produce better results than several well-known contemporary methods such as CQPSO, CS, CHBMA, EPO, and FA. That is, the proposed algorithm dominated the state-of-the-art population-based thresholding method in term of function fitness value, optimal threshold value, SSIM value, PSNR value, computation time. The optimal solution prediction technology from the proposed algorithm can also be used in other population-based color image segmentation methods.

The future work is on how to apply it to video compressing for effective transmission, wildlife monitoring with wireless multimedia sensor networks, satellite image enhancement and satellite image denoising.

Acknowledgments: Supported by Natural Science Foundation of China (61702118), the Chinese Scholarship Council (CSC), Overseas Master Project of Department of Finance of Guangdong Province (Document No.: YUE-CAI-KE-JIAO [2022]184), Department of Education of Guangdong Province (2022ZDJS013).

Declaration of Interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

 doi:10.20944/preprints202406.1791.v1

Notation:

L	The gray levels belonging to [0, L–1]
$h(r)$	Observed frequency of gray-level r, where r =0, 1,…, L–1
N	The sum number of pixels in image
p_r	The ratio of the observed frequency of the rth gray-level to N, where r =0, 1,…, L–1
M	The number of thresholds $1 \le M \le L-1$
t_j	jth thresholds, where j = 1,2,…,M.
E_k	Entropy of kth class of image divided by the thresholds, where $k = C_0, C_1, C_2, \ldots, C_M$.
$f(\bullet)$	Objective function of event •
w_k	Cumulated frequency of p_r, where $k = C_0, C_1, C_2, \ldots, C_M$ and r =0, 1,…, L–1
$Rand(1)$	A random number from the uniform distribution [0, 1]
	The breeding coefficient
D	The dimensionality of an individual and is the sum of the number of thresholds M and 1
X_i	An arbitrary individual from the population
$BestX$	The individual with the maximum fitness is selected as $BestX$
X_{id}	The position status of an arbitrary individual X_i in d^{th} dimension, i.e. X_{id}, is generated from the uniform distribution in the closed interval [(d-1)×floor(L/D), d×floor(L/D)], i=1, 2, …, *PopulationSize*.
$PVector[n]$	Prediction vector
$TopSize$	Number of elite individuals

References

1. Jiang, Y., Yang, Z., Hao, Z., Wang, Y., and He, H.: A cooperative honey bee mating algorithm and its application in multi-threshold image segmentation, in Editor (Ed.) A cooperative honey bee mating algorithm and its application in multi-threshold image segmentation (2014,edn.), pp. 1579-1585
2. Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., and Osuna, V.: A Multilevel Thresholding algorithm using electromagnetism optimization, NEUROCOMPUTING, 2014, 139, pp. 357-381
3. Jiang, Y., Tsai, P., Hao, Z., and Cao, L.: Automatic multilevel thresholding for image segmentation using stratified sampling and Tabu Search, SOFT COMPUT, 2015, 19, (9), pp. 2605-2617
4. Gao, H., Xu, W., Sun, J., and Tang, Y.: Multilevel Thresholding for Image Segmentation Through an Improved Quantum-Behaved Particle Swarm Algorithm, IEEE T INSTRUM MEAS, 2010, 59, (4), pp. 934-946
5. Dey, S., Saha, I., Bhattacharyya, S., and Maulik, U.: Multi-level thresholding using quantum inspired meta-heuristics, KNOWL-BASED SYST, 2014, 67, (3), pp. 373-400
6. Aziz, M.A.E., Ewees, A.A., and Hassanien, A.E.: Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, EXPERT SYST APPL, 2017, 83, pp. 242-256
7. Kumar, S., Pant, M., Kumar, M., and Dutt, A.: Colour image segmentation with histogram and homogeneity histogram difference using evolutionary algorithms, INT J MACH LEARN CYB, 2018, 9, (1), pp. 163-183

 doi:10.20944/preprints202406.1791.v1

8. Zhao, F., Liu, H., Fan, J., Chen, C.W., Lan, R., and Li, N.: Intuitionistic fuzzy set approach to multi-objective evolutionary clustering with multiple spatial information for image segmentation, NEUROCOMPUTING, 2018
9. Naidu, M.S.R., Rajesh Kumar, P., and Chiranjeevi, K.: Shannon and Fuzzy entropy based evolutionary image thresholding for image segmentation, Alexandria Engineering Journal, 2018, 57, (3), pp. 1643-1655
10. Liu, Z., Xiang, B., Song, Y., Lu, H., and Liu, Q.: An Improved Unsupervised Image Segmentation Method Based on Multi-objective Particle Swarm Optimization Clustering Algorithm, Computers, Materials & Continua, 2019, 58, (2), pp. 451-461
11. Bhandari, A.K., Kumar, I.V., and Srinivas, K.: Cuttlefish Algorithm-Based Multilevel 3-D Otsu Function for Color Image Segmentation, IEEE T INSTRUM MEAS, 2020, 69, (5), pp. 1871-1880
12. Xing, Z.: An improved emperor penguin optimization based multilevel thresholding for color image segmentation, KNOWL-BASED SYST, 2020, pp. 105570
13. Horng, M.: A multilevel image thresholding using the honey bee mating optimization, APPL MATH COMPUT, 2010, 215, (9), pp. 3302-3310
14. YunZhi, J., ZhiFeng, H., YuShan, Z., Han, H., YingLong, W., and HuoJiao, H.: Bayesian Forecasting Evolutionary Algorithm, Chinese Journal of Computers, 2014, 37, (8)
15. Jiang, Y., Tsai, P., Yeh, W., and Cao, L.: A honey-bee-mating based algorithm for multilevel image segmentation using Bayesian theorem, APPL SOFT COMPUT, 2017, 52, pp. 1181-1190
16. Agrawal, S., Panda, R., Bhuyan, S., and Panigrahi, B.K.: Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, SWARM EVOL COMPUT, 2013, 11, pp. 16-30
17. Dhiman, G., and Kumar, V.: Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, KNOWL-BASED SYST, 2018, 159, pp. 20-50
18. Naidu, M.S.R., Rajesh Kumar, P., and Chiranjeevi, K.: Shannon and Fuzzy entropy based evolutionary image thresholding for image segmentation, Alexandria Engineering Journal, 2018, 57, (3), pp. 1643-1655
19. Rajinikanth, V., and Couceiro, M.S.: RGB Histogram Based Color Image Segmentation Using Firefly Algorithm, Procedia Computer Science, 2015, 46, pp. 1449-1457
20. He, L., and Huang, S.: Modified firefly algorithm based multilevel thresholding for color image segmentation, NEUROCOMPUTING, 2017, 240, pp. 152-174