Evolutionary Intelligence (2019) 12:445–467
https://doi.org/10.1007/s12065-019-00238-1

RESEARCH PAPER

Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding

Falguni Chakraborty[1] · Provas Kumar Roy[2] · Debashis Nandi[1]

Received: 27 September 2018 / Revised: 22 March 2019 / Accepted: 2 May 2019 / Published online: 21 May 2019

Abstract
This paper presents an improved elephant herding optimization (IEHO) to solve the multilevel image thresholding problem for image segmentation by introducing oppositional-based learning (OBL) and dynamic cauchy mutation (DCM). OBL accelerates the convergence rate and enhances the performance of standard EHO whereas DCM mitigates the premature convergence. The suggested optimization approach maximizes two popular objective functions: 'Kapur's entropy' and 'between-class variance' to estimate optimized threshold values for segmentation of the image. The performance of the proposed technique is verified on a set of test images taken from the benchmark Berkeley segmentation dataset. The results are analyzed and compared with conventional EHO and other four popular recent metaheuristic algorithms namely cuckoo search, artificial bee colony, bat algorithm, particle swarm optimization and one classical method named dynamic programming found from the literature. Experimental results show that the proposed IEHO provides promising performance compared to other methods in view of optimized fitness value, peak signal-to-noise ratio, structure similarity index and feature similarity index. The suggested algorithm also has better convergence than the other methods taken into consideration.

Keywords Multi-level thresholding · Nature inspired optimization · Elephant herding optimization · Image segmentation · Opposition based learning · Cauchy mutation

1 Introduction

Image segmentation is an important step in pattern recognition, computer vision and image processing. It is a process of partitioning different non-overlapping regions of an image. Many segmentation techniques like threshold based, edge based, region based, graph cut based, connectivity-preserving relaxation techniques, etc. have been proposed to segment different objects of an image. Threshold-based segmentation among them has become popular to the research community because of its simplicity, accuracy, and robustness. However, the primary challenge for this category of segmentation is to find out the best optimized threshold values to partition the different regions with sufficient accuracy. In most of the cases, the traditional optimization algorithms are inefficient to solve major real-world problems because of their failure to search the best set of optimized threshold values. Nature inspired metaheuristic optimization algorithms are thought recently to be a promising solution in this regard.

The real-world problems are somehow associated with some optimization problems. Hence, optimization has got tremendous importance as an area of research in the course of time. However, it has now become a large area of mathematics and encompasses a lot of techniques to solve different types of problems. These techniques are broadly categorized into classical algorithms [1, 2] and evolutionary algorithms [3, 4]. The classical algorithms are particularly gradient-based, and the optimized values are estimated by evaluating the derivatives of the objective functions. The evolutionary algorithms are metaheuristic and inspired by the nature or biological behavior of the living entities. Since the objective functions of many real-world problems are multimodal, the gradient-based classical methods fail to assure global or near global solutions. There is a high chance that the solutions will stick at the local optima depending on the assumption of the initial solutions. Moreover, the derivative based algorithms unsuitable for those problems whose objective

✉ Falguni Chakraborty
falguni.durgapur@gmail.com

1 National Institute of Technology, Mahatma Gandhi Avenue, Durgapur, West Bengal, India

2 Kalyani Government Engineering College, Kalyani, West Bengal, India

functions are not differentiable. These factors lead us to search for new algorithms for finding global or near global solutions for a given problem. Evolutionary algorithms have become potential solutions for these cases.

Recently, a wide range of nature inspired metaheuristic evolutionary algorithms have been proposed and applied to solve different types of large-scale optimization problems. Some examples of such algorithms are: genetic algorithm (GA) [5–7], biogeography based optimization (BBO) [8], particle swarm optimization (PSO) [9, 10], artificial bee colony (ABC) [11], modified ABC [12], differential evolution (DE) [13], bacterial foraging optimization (BFO) [14], ant colony optimization (ACO) [15], cuckoo search (CS) [16], honey bee mating optimization (HBMO) [17], social spider optimization (SSO) [18], flower pollination [18], BAT algorithm [19] etc. These algorithms are successfully applied to solve the complicated and challenging engineering problems in different fields. Some good references in this regard are Bhandari et al. [12], a multilevel thresholding algorithm for segmentation of gray-level satellite image using a modified artificial bee colony algorithm (MABC), Ouadfel et al. [18], social spiders optimization (SSO) and flower pollination algorithm for multilevel image thresholding, Agrawal et al. [16], Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, Bakhshali et al. [20], segmentation of color lip images using BFO, Khairuzzaman et al. [21], multilevel image thresholding using grey wolf optimizer, Aziz et al. [22], Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, etc. Moreover, in line with the no free-lunch theorem [23], a particular metaheuristic algorithm may not be able to produce the best result in the solution of all types of problems. This leads us to develop and/or search a metaheuristic algorithm or enhance the performance of the existing algorithm by various hybrid or improved version. Some such hybrid algorithms are: enhancing particle swarm optimization using generalized opposition-based learning [24]; opposition-based artificial bee colony with dynamic Cauchy mutation [25]; opposition-based kill herd algorithm with Cauchy mutation and position clamping [26]; opposition-based differential evolution [27] etc.

EHO is newly developed metaheuristic algorithm, proposed by Wang et al. [28] This algorithm has been inspired by the herding behaviour of the elephant in nature and is applied to improve the solutions of many real-life problems. Meena et al. [29] have used improved elephant herding optimization for multi-objective DER accommodation in distribution systems. Tuba et al. [30] have applied EHO to obtain optimal parameters for the support vector machine to determine the exact erythemato-squamous diseases. However, though EHO has shown its potential performance with better results in many cases; sometimes premature stagnation at local optima makes it inferior with respect to other contemporary evolutionary algorithms. To combat it, we propose a variant of EHO by incorporating opposition-based learning (OBL) [31] and Dynamic Cauchy Mutation (DCM) [32–34] to EHO and use it to obtain the improved optimal threshold in multilevel image thresholding with higher speed.

The opposition based learning (OBL) was first proposed by Tizhoosh [31]. The key concept of OBL is to search for better candidate solutions by the simultaneous evolution of an estimate and its corresponding opposite estimate which is closer to the global optimum. According to central opposition theorem [35],the probability that the opposite candidate solution is closer to the global optimum is greater than the probability of a second random presumption. Many researchers have utilized the concept of OBL in combination with different soft computing algorithms. These include particle swarm optimization (PSO), artificial bee colony (ABC), Krill Herd algorithm (KH) and Differential Evolution (DE) to solve the real-life complex optimization problems. In this paper, OBL and DCM have been incorporated in the conventional EHO to accelerate the convergence rate as well as to enrich the performance of standard EHO in terms of the optimized threshold values.

We organize the article as follows: Sect. 2 presents the basic EHO and OBL theory, Sect. 3 proposes the improved EHO. Section 4 represents the mathematical model for image segmentation. Section 5 demonstrates the application of the proposed algorithm to solve the image thresholding problem. Section 6 describes the simulation results of the proposed improved EHO on multilevel thresholding in image segmentation. Finally, the conclusions are drawn in Sect. 7.

2 Elephant herding and opposition based learning algorithms

This section presents a brief overview of elephant herding and opposition based learning algorithm.

2.1 Elephant herding optimization (EHO) algorithm

In Wang et al. [28] developed a new metaheuristic algorithm called elephant herding optimization (EHO) algorithm. The EHO algorithm is developed from the natural behaviour of elephant herding. An elephant group is composed of a number of clans headed by a matriarch. A clan consists of females with their calves. Female members prefer to live with family members whereas the male members prefer to live a nomadic and solitary life. Therefore, they will gradually become independent of their families until they break away the relationship with their families completely to either roam alone or find a small group of male elephants.

Wang et al. have modelled this herding behaviour of elephant into clan updating and separation operator to solve an optimization problem. For solving an optimization problem using this herding behaviour of the elephant, the following assumptions are considered.

- Each elephant group is composed of some number of clans and each clan consists of a fixed number of elephants under the headship of a matriarch
- In each generation, a fixed number of male elephants live away from the clan and they are considered as minimum fitness value (for the maximization problem).
- The matriarchs are represented by the maximum fitness value.

2.1.1 Clan updating operator

As matriarchs are the leaders of the elephant group, all the members of a particular clan c_i are influenced by the matriarch of the clan c_i, so in each generation the next position of the member's *j*th elephant of the clan c_i is calculated as

$$x_{new,c_i,j} = x_{c_i,j} + \alpha \times (x_{best,c_i} - x_{c_i,j}) \times r \tag{1}$$

where $x_{best,ci}$ is the fittest elephant of the clan is c_i, $x_{new,ci,j}$ represents the new updated position of *j*th elephant in clan c_i respectively. Matriarch influence to the *j*th elephant of the clan c_i is determined by the control parameter $\alpha \in [0, 1]$ and $r \in [0, 1]$, is a random number taken from a set of uniformly distributed numbers. The next updated position of fittest elephant for $x_{ci,j} = x_{best,ci}$ is calculated as follows

$$x_{new,c_i,j} = \beta \times x_{center,c_i} \tag{2}$$

where $\beta \in [0, 1]$ is a control parameter which determines the influence factor of $x_{center,ci}$ on $x_{new,ci,j}$. The $x_{center,ci}$ of the clan c_i can be determined by the following equation.

$$x_{center,c_i,d} = \frac{1}{n_{c_i}} \times \sum_{n_{c_i}}^{j=1} x_{c_i} d \tag{3}$$

where n_{c_i} is the number of elephants of each clan; $1 \leq d \leq D$ represents the *d*th dimension and D is the total dimension.

2.1.2 Separating operator

In the elephant group, young male elephants prefer to leave alone so, they leave their family members. This characteristic is implemented in this optimization technique as separating operator. In each generation worst elephant of each clan updates its position by the following.

$$x_{wrost,c_i} = x_{min} + (x_{max} - x_{min}) \times rand \tag{4}$$

where x_{worst,c_i} is the worst elephant of the clan c_i of elephant group x_{max}, x_{min} are the maximum and minimum value of the search space and $rand \in [0, 1]$ follows the uniform distribution.

2.2 Opposition-based learning (OBL)

Generally, the initial population for all the evolutionary algorithms is generated randomly and gradually they reach to the optimal solution in subsequent iterations and stop at the predefined condition. The convergence time of the algorithms are linked to the distances of these initial guesses from the optimal solution. If the selection of the initial solution is closer to the optimal solution then it converges quickly, otherwise, it takes a longer time to converge. Opposition based learning (OBL) [29] is one of the efficient concepts to improve the initial solution by simultaneously evaluating the current candidate solution and its opposite solution and choose the more fitted one as the initial solution. The underlying concept is that as per probability theory any predicted solution is 0.5 times far away from the actual solution than its contrary solution. The method is useful not only for starting population but also for each iteration to improve the final solution.

The concept of OBL in optimization problem is based on simultaneously evaluation of current candidate solution and its opposite solution. Some definitions related to OBL are given below:

2.2.1 Opposite number

Opposite number ($\bar{x}$) of a real number (x) is express as $\bar{x} = a + b - x$, where $x \in [a, b]$. The same concept may be applied for generating opposite number in multidimensional problem.

2.2.2 Opposite point

Let $P = (x_1, x_2, \dots, x_D)$ be a point in D-dimensional space, where $x_i \in R$ and $x_i \in [a_i, b_i]$ and $\{i = 1, 2, \dots, D\}$. The opposition point $P_o(x_1, x_2, \dots, x_D)$ can be defined as

$$\bar{x} = a_i + b_i - x_i \tag{5}$$

The concept of this opposition based optimization by using the idea of opposite point is defined as follows:

Let $P = (x_1, x_2, \dots, x_D)$ be the candidate solutions in *D*-dimensional problem space and *f(P)* be the fitness function for measuring the candidate fitness. The point P_o is the opposite point of the point *P*. In maximization problem,the point *P* is replaced by the point P_o if $f(P_o) \geq f(P)$. Hence, a point and its opposite point are evaluated at the same time and choose the fittest one.

3 Proposed opposition based EHO with DCM

The basic operations of all evolutionary algorithms are divided into two parts, initialization and produced a new population in the subsequent generations. In the proposed algorithm we will enhance these two sections by embedding the concept of opposition based learning and dynamic Cauchy mutation into the standard EHO to improve the performance of the EHO as well as convergence speed. Details of the proposed algorithm ae shown in Fig. 1.

3.1 Opposition based initial population

The literature review of evolutionary algorithms reveals that almost all evolutionary algorithms are started with the randomly generated initial population without prior information of the problem area. Here, the concept of OBL can be useful to create fitter starting population without prior knowledge of the problem field. In the proposed algorithm a population is produced by the conventional random number generator, and then the opposite population is produced and combined with the original population. Finally, the initial population is taken by selecting the best subpopulation depending on the fitness value.

3.2 Opposition based generation by jumping probability

The similar concept may be used in each iteration to improve the solution by forcefully changing the current population to its opposite population by using generation jumping (J_r) concept, which may be fitter than the old population. After generating a new population by the IEHO in each iteration, the opposite population is formed and combined with the old population. Then we select the fittest population from the merged population as a new population for the next iteration.

According to the literature, generation jumping (J_r) during exploration is more desirable than during exploitation [36]. In our segmentation problem if we set the value of jumping rate (J_r) to 0.4 or 0.2 we see premature convergence for some images. We observed that if we set the value between 0.2 and 0.4, we get desired results for a large number of image data.

3.3 Dynamic Cauchy Mutation (DCM)

Various mutation operators are proposed in the literature of evolutionary optimization to improve the performance by avoiding premature convergence. Among them, Gaussian and Cauchy distribution have become popular. Compare to the Gaussian probability distribution, Cauchy probability distribution has more possibility to escape from local optima because of its longer tail probability distribution function. This motivates us to used Cauchy probability distribution as a mutation operator to enrich the performance of the conventional EHO.

In the conventional EHO, an elephant group is composed of a number of clans headed by a matriarch (global best). The other members of the clan update them by the influence of matriarch to reach better position. Hence matriarch can guide when the other member of the clan tends to be trapped.

In this algorithm, the DCM is applied on matriarch to enhance the performance of EHO.We apply the DCM operator on the matriarch as follows:

$$x_{c_i,M} = x_{c_i,M} + \delta \times CM() \tag{6}$$

where $x_{c_i,M}$ is the matriarch of clan c_i; δ is the dynamic weight and $CM()$ is the random number generator by the Cauchy probability distribution.

Dynamic weight δ is calculated as follows

$$\delta = \delta_0 + \frac{MI - I}{I} \times \delta_1 \tag{7}$$

where $\delta_0 = 0.01$; MI is the maximum number of iteration; I is the current iteration; δ_1 is calculate as follows

$$\delta_1 = \frac{N_{max} - N_{min}}{1000} \times \delta_1 \tag{8}$$

where N_{max}, N_{min} are the maximum and minimum value of problem domain.

4 Mathematical model of thresholding problem

Thresholding is a method of dividing an image into dissimilar regions based on the intensity level (L).In bi-level thresholding, the background and foreground of an image may be separated by a threshold value by the following rules.

$$\begin{aligned} &R_1 \leftarrow P \quad && if \quad 0 \le P < T \\ &R_2 \leftarrow P \quad && if \quad T \le P < L-1 \end{aligned} \tag{9}$$

where threshold value T divides the image into two regions and is the one of the value of the pixel in L-level gray scale image. Multiple threshold value require more than one threshold values, which divided the whole image into multiple regions. The idea of bi-level thresholding can be express to multilevel thresholding techniques also by the following rules:

$$\begin{aligned} &R_1 \leftarrow P \quad if\ 0 \le P < T_1 \\ &R_2 \leftarrow P \quad if\ T_1 \le P < T_2 \\ &R_{n-1} \leftarrow P \quad if\ T_{n-1} \le P < T_n \\ &R_n \leftarrow P \quad if\ T_n \le P < L-1 \end{aligned} \tag{10}$$

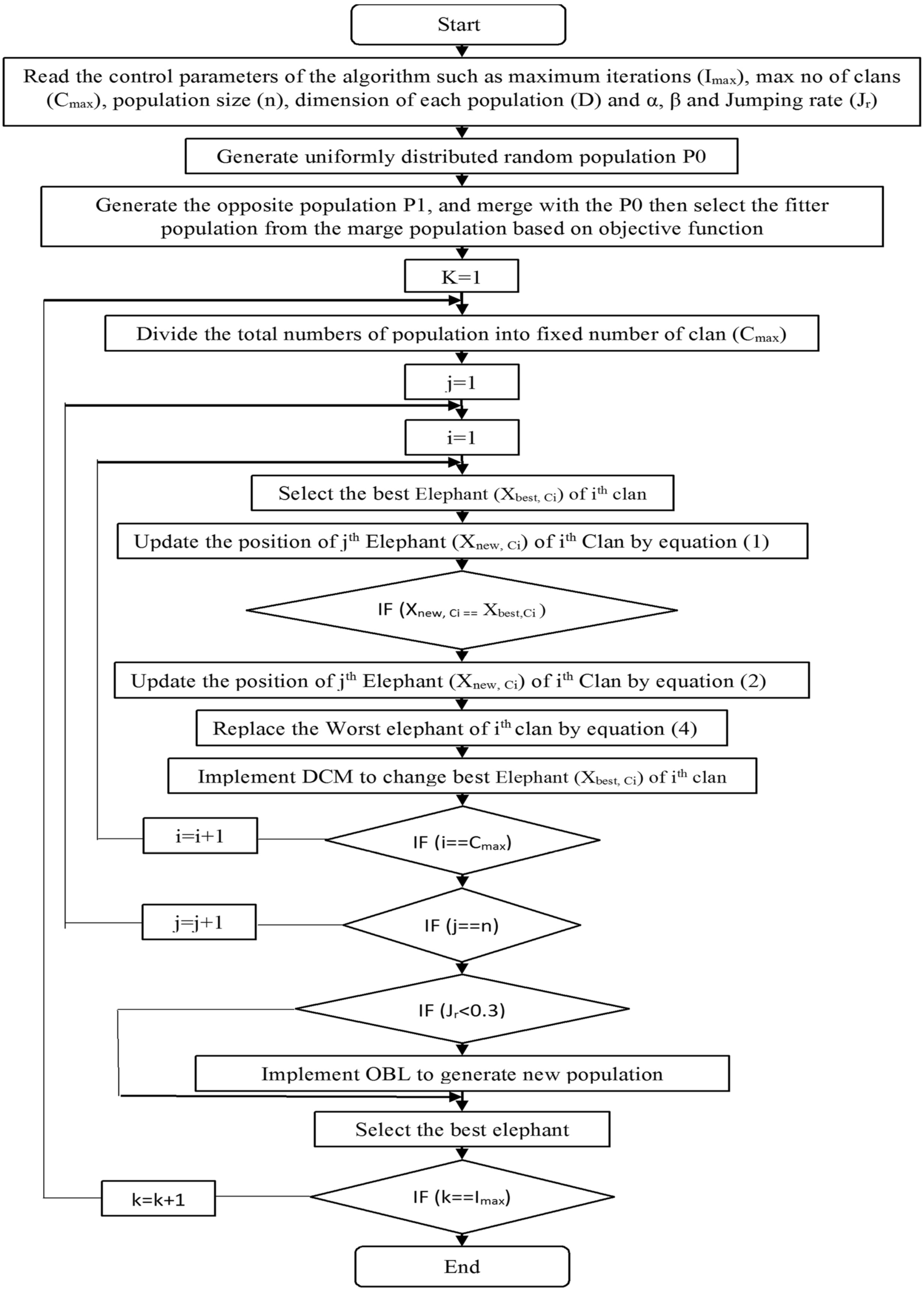


Fig. 1 Details of improved EHO(IEHO) algorithm

where $T_1 < T_2 < \cdots < T_{n-1} < T_n$ are the threshold value.

Let assume an L number of grey image in the range $\{0, 1, 2, \dots .(L-1)\}$ with $n+1$ number of objects or regions. $\{h(1), h(2), \dots, h(L-1)\}$ be the grey level frequency of gray level $\{0, 1, 2, \dots .(L-1)\}$, N be the total number of pixel in that image. Where it is required to find n number of threshold values by optimizing one (or more) objective function(s). If an objective function is $f(.)$, the optimal threshold values $T_1^*, T_2^*, \dots, T_n^*$ can be computed as follows:

$$(T_1^*, T_2^*, \dots T_n^*) = argmax(f(T_1, T_2, \dots, T_n)) \tag{11}$$

The choice of objective functions is one of the important issues in finding out the optimal threshold values. Otsu [37], Kapur's entropy [38] method are some examples of popular objective functions which are widely used in image segmentation problems discussed in the next subsection.

4.1 Otsu's method

This is one of the most popular methods, proposed by Otsu's [37], for both bi-level and multiple thresholding, it is based on finding the optimal thresholding by maximizing the between-class variance of the segmented region which can be defined as the sum of sigma functions of each region by the equation given below:

$$f(t) = \sigma_1 + \sigma_2 \tag{12}$$

where

$$\sigma_1 = \omega_0(\mu_0 - \mu_T)^2 \; and \; \sigma_2 = \omega_1(\mu_1 - \mu_T)^2 \tag{13}$$

where μ_T in Eq. 13 represents the mean intensity of the whole image, and for bi-level thresholding. Mean of each class can be defined as

$$\mu_0 = \sum_{i=0}^{t-1} \frac{ip_i}{\omega_0} \; and \; \mu_1 = \sum_{i=t}^{L-1} \frac{ip_i}{\omega_1} \tag{14}$$

The optimal threshold can be obtained by maximizing between-class variance.

$$(t^*) = argmax(f(t)) \tag{15}$$

This method can be express for multilevel thresholding problem by the following function

$$f(t) = \sum_{i=0}^{m} \sigma_i \tag{16}$$

The sigma function can be express through Eq. 17

$$\begin{aligned} &\sigma_1 = \omega_1(\mu_0 - \mu_T)^2, \; \sigma_2 = \omega_2(\mu_1 - \mu_T)^2, \\ &\sigma_j = \omega_j(\mu_j - \mu_T)^2, \; \sigma_m = \omega_m(\mu_m - \mu_T)^2 \end{aligned} \tag{17}$$

And mean level van be defined as

$$\begin{aligned} &\mu_0 = \sum_{i=0}^{t_1-1} \frac{ip_i}{\omega_i}, \; \mu_1 = \sum_{i=t_1}^{t_2-1} \frac{ip_i}{\omega_i}, \; \mu_j = \sum_{i=t_j+1}^{L-1} \frac{ip_i}{\omega_i}, \\ &\mu_m = \sum_{i=t_m}^{t_j+1-1} \frac{ip_i}{\omega_i} \end{aligned} \tag{18}$$

Optimal threshold value can be obtained by maxizing the objective function of the Eq. 19

$$(t^*) = argmax\left(\sum_{i=0}^{m} \sigma_i\right) \tag{19}$$

4.2 Kapur's entropy method

The most popular and widely used entropy-based method is Kapur's entropy-based [38] thresholding. It describes the method to maximize the entropy of the segmented histogram in order that each segmented section has a more centralized distribution [38].

Optimal threshold for bi-level thresholding by Kapur's entropy can be defined as maximizing the following function

$$f(t) = H_0 + H_1 \tag{20}$$

where

$$H_0 = -\sum_{i=0}^{t-1} \frac{p_i}{\omega_0} ln \frac{p_i}{\omega_0} \text{ and } \omega_0 = \sum_{i=0}^{t-1} p_i$$

$$and \; H_1 = -\sum_{i=t}^{L-1} \frac{p_i}{\omega_1} ln \frac{p_i}{\omega_1} \text{ and } \omega_1 = \sum_{i=0}^{t-1} p_i$$

This entropy method can also be express for multilevel thresholding by maximizing the following equation

$$f([t_1.t_2, t_3 \dots t_m]) = H_0 + H_1 + H_2 + \cdots + H_n \tag{21}$$

where $t_1 < t_2 < t_3 \cdots < t_m$ and

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} ln \frac{p_i}{\omega_0} \text{ and } \omega_0 = \sum_{i=0}^{t_1-1} p_i$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} ln \frac{p_i}{\omega_1} \text{ and } \omega_1 = \sum_{i=t_1}^{t_2-1} p_i$$

$$H_2 = -\sum_{i=t_j}^{t_{j+1}-1} \frac{p_i}{\omega_2} ln \frac{p_i}{\omega_2} \text{ and } \omega_2 = \sum_{i=t_j}^{t_{j+1}-1} p_i$$

$$H_n = -\sum_{i=t_n}^{L-1} \frac{p_i}{\omega_n} ln \frac{p_i}{\omega_n} \text{ and } \omega_m = \sum_{i=t_n}^{L-1} p_i$$

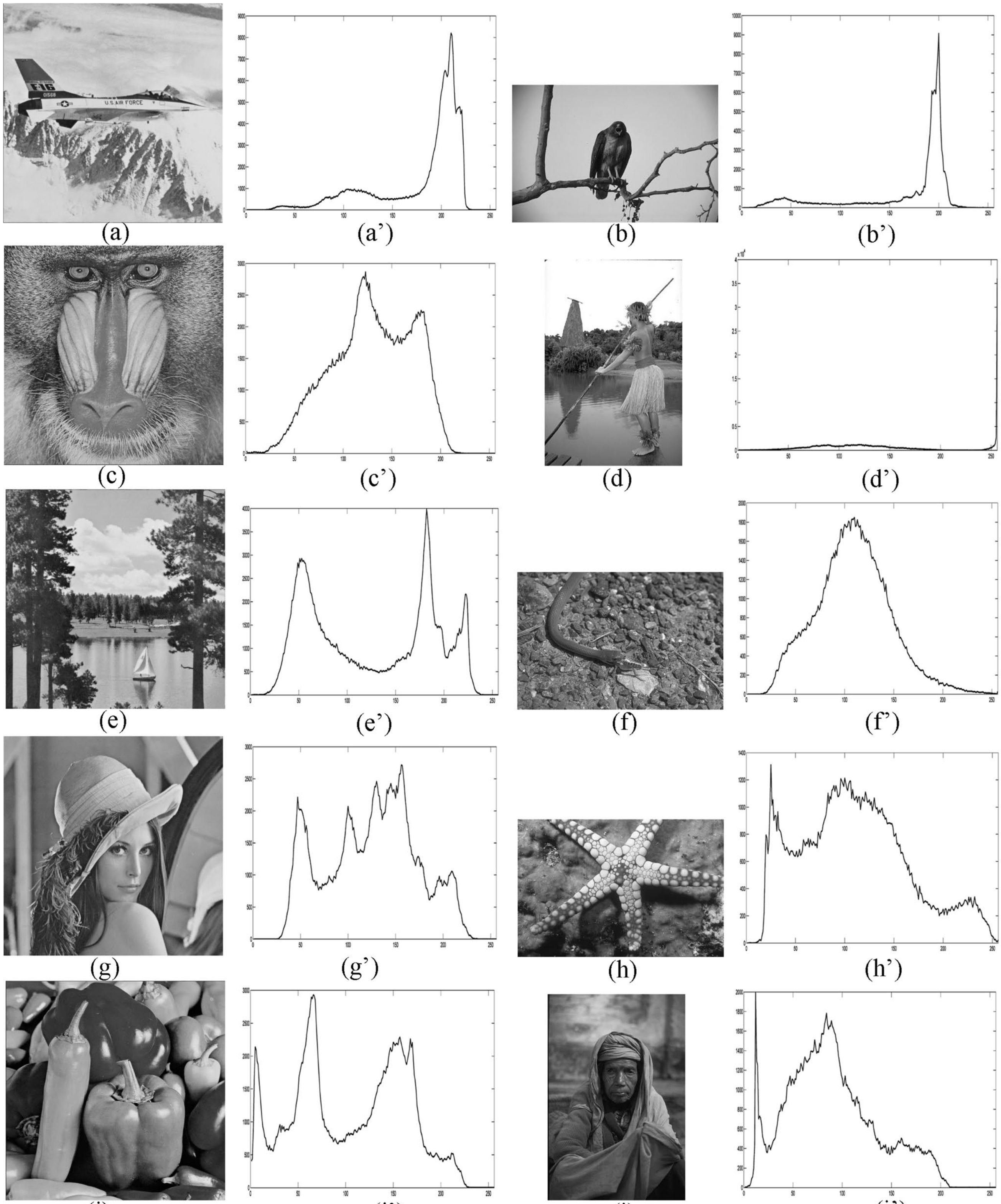


Fig. 2 (**a**–**j**) original images and (**a'**–**j'**) are the histogram of the images respectively

Table 1 Parameters used for EHO, ABC, CS, BAT and, PSO

Algorithms	Parameter	Value	Algorithms	Parameters	Value
EHO	Population size	30	ABC	Swam size	30
	No. of iterations	100		No. of iterations	100
	Matriarch influence parameter (α)	0.9		Lower bound and upper bound	1256
	Influence factor of clan canter(β)	0.4		Value of Fi (ϕ)	[0, 1]
CS	Number of nests	30	PSO	Population size	30
	No. of iterations	100		No. of iterations	100
	Step size	1		Maximum Inertia weight	0.4
	Mutation probability value	0.25		Minimum Inertia weight	0.2
	Scale factor	1.5		Maximum velocity	+ 1.0
				Minimum velocity	− 1.0
				Cognitive coefficient(c1)	1.429
				Cognitive coefficient(c2)	1.429
BAT	Number of Bat	30	BAT	Minimum and maximum Pulse emission	0, 1
	No. of iterations	100		Minimum and maximum Loudness	1, 2
	Maximum frequency	1.5		Minimum and Maximum wavelength	0.9, 0.99
	Minimum frequency	0			

5 Improved EHO and multilevel image thresholding

The IEHO has been used to find the optimal threshold values for multilevel image thresholding of the images taken from well-known benchmark dataset. The steps for implementing IEHO for multilevel image thresholding are as follows:

Step 1 Select the initial population based on the concept of OBL of size n, each of the member of the population is of dimension D
Step 2 Divide the total population into a fixed number of clans.A matriarch controls every clan. Since it is a maximization problem, the matriarch gives the highest fitness value. For each clan c_i, calculate the fitness function of the jth elephant by the Kapur's or Otsu' s method
Step 3 In each generation, the next position of the jth member of each clan c_i is updated by the Eqs. 1 and 2.
Step 4 The worst member of each clan is updated its position by the Eq. 4.
Step 5 Implement the OBL based on jumping rate (J_r).
Step 6 Implement the DCM to update matriarch
Step 7 Select the best elephant as the best threshold value from the group in each iteration.
Step 8 Repeat Step-2 to Step-7 until the maximum iteration is reached.

6 Experiment

6.1 Experiment and setup

This section describes the computational environment used for simulation and experiment of the proposed algorithm; the results are analysis by defining some well-known quality metrics; the stability and the performance of the proposed algorithm in comparison to the other popular algorithms.

Simulation and experiments are carried out in a PC with 2.30 GHz CPU and 4.00 GB RAM in Windows 8 environment. Tests are performed on more than fifty different images collected from the database of Berkeley Segmentation Dataset, out of which the results of ten images are shown in Fig. 2. The same number of iterations and stopping criteria are used for all algorithms to assess the performance without any bias. The parameters used in simulations are given in Table 1. The performance of the proposed algorithm is compared with five recently developed evolutionary algorithms namely, CS, ABC, BAT and, PSO and one classical method DP. Kapur's entropy and Otsu's between-class variance are used as the objective fnction for calculating the threshols value of images for all metaheuristic algorithm, In case of DP Modified Otsu Criterion [2] proposed by the Mohamed H. Merzban, Mahmoud Elbayoumi is considered for calculating the threshold values.

Table 2 Threshold values for ten images in different optimization techniques taking Kapur's entropy as optimization function

Threshold Values							
Image	K	IEHO	EHO	ABC	CS	PSO	BAT
Airplane	5	65 95 127 157 187	65 95 127 157 187	65 95 127 157 187	65 95 127 157 187	50 77 110 149 185	65 95 127 157 187
	6	49 74 102 131 160 189	49 74 102 131 160 189	50 74 100 131 160 188	48 74 102 131 159 188	47 74 101 131 156 187	49 74 101 131 161 189
	7	48 71 96 123 148 173 195	47 71 96 122 147 174 198	47 70 95 120 142 167 191	47 70 96 121 148 172 192	47 71 96 124 151 177 197	47 71 96 120 145 170 193
Baboon	5	23 58 95 132 168	23 58 95 132 168	23 57 96 134 170	23 55 93 132 168	23 58 95 132 168	23 54 93 131 167
	6	23 48 79 110 142 174	23 48 79 110 142 174	23 53 84 114 144 178	23 49 80 111 143 175	23 50 82 113 147 184	23 49 80 111 141 173
	7	23 48 76 105 133 161 189	23 48 77 106 135 162 190	23 48 79 108 134 162 191	23 48 78 104 132 161 189	23 49 78 107 135 163 191	23 48 78 107 134 162 190
Bird	5	60 103 146 184 212	60 103 146 184 212	60 103 146 184 212	58 100 144 184 212	59 102 145 182 212	61 103 146 184 212
	6	51 85 118 152 184 212	51 85 118 152 184 212	51 85 118 152 184 212	54 87 123 153 184 212	51 82 115 151 184 212	51 83 115 147 184 212
	7	42 71 101 130 159 187 212	42 71 101 129 159 186 212	44 75 102 128 158 184 212	48 82 113 152 184 211 233	51 85 118 152 184 211 228	39 67 98 128 158 185 212
Fishing	5	48 100 150 197 242	48 100 150 197 242	48 100 150 197 242	48 101 150 197 242	48 100 150 197 242	46 99 150 197 242
	6	38 73 113 154 198 242	38 73 113 154 199 241	38 73 113 154 198 242	34 71 112 154 198 242	38 73 113 154 199 241	35 71 110 155 198 242
	7	34 68 102 136 169 204 241	34 67 102 135 169 204 241	34 68 103 137 169 204 241	38 71 107 143 176 204 241	34 68 102 137 169 204 241	38 71 107 143 176 204 241
Lake	5	66 100 134 167 198	66 97 131 165 198	67 96 128 163 196	65 103 136 167 198	68 109 141 168 196	66 99 134 167 198
	6	39 72 105 137 168 198	39 72 105 137 168 198	41 72 104 136 168 198	42 76 108 139 168 198	39 72 104 136 168 198	42 72 105 137 168 198
	7	15 42 73 105 137 168 198	15 42 74 106 138 168 198	15 44 71 101 130 164 196	15 44 69 104 136 166 198	15 45 76 107 138 168 198	15 42 72 102 135 167 199
Lena	5	63 94 127 163 194	63 94 127 163 194	63 94 127 163 194	64 94 127 163 195	63 94 127 161 192	98 166 237 28 121
	6	61 89 116 142 168 196	61 89 116 142 168 196	61 89 116 141 168 197	62 88 114 139 166 197	61 89 116 142 169 198	61 89 115 142 168 196
	7	60 86 112 136 160 181 203	60 86 112 136 159 180 202	64 88 113 137 160 179 201	63 87 111 135 156 179 204	61 88 114 139 164 189 216	62 88 113 136 160 181 203
Peppers	5	42 75 110 144 181	42 75 110 144 181	41 75 110 144 181	41 75 112 146 181	41 75 110 144 181	42 75 109 144 181
	6	42 75 108 140 174 199	42 75 107 139 174 199	42 73 106 138 174 199	43 74 108 137 174 201	36 74 107 141 174 199	42 74 106 138 174 198
	7	12 50 77 109 141 174 199	26 50 78 111 142 174 199	25 50 77 107 142 175 200	22 49 76 107 141 174 199	25 50 77 109 141 174 198	27 51 79 109 143 175 199
Snake	5	65 104 143 180 219	65 104 143 180 219	65 103 143 180 219	65 105 144 181 219	65 104 143 180 219	65 103 143 180 219
	6	55 87 122 162 187 221	55 87 122 155 187 221	55 88 125 157 187 221	58 91 126 160 195 224	58 92 126 159 191 221	53 86 122 155 187 221
	7	51 80 110 140 168 195 224	51 80 110 134 167 195 224	50 78 110 139 167 195 225	53 79 109 138 165 192 221	52 81 112 142 170 198 224	52 82 110 139 166 195 224
Starfish	5	56 94 133 172 211	56 94 133 172 211	51 87 131 169 210	57 94 133 172 210	56 95 135 176 213	56 94 133 172 211
	6	49 81 115 149 181 215	49 81 115 149 181 215	49 81 115 149 181 216	50 82 116 149 182 216	49 82 116 151 182 215	49 82 116 149 182 216
	7	47 77 107 136 165 193 222	46 76 105 134 163 191 221	46 76 107 136 166 194 223	45 74 105 136 165 193 222	45 75 106 135 165 195 223	46 76 106 136 166 194 222
Women	5	42 73 105 136 170	42 73 105 136 170	42 74 105 136 170	42 73 106 137 171	42 73 105 136 170	43 74 106 138 172
	6	39 67 95 122 150 178	39 67 95 122 150 178	39 67 96 122 151 178	39 66 95 122 151 179	39 67 96 123 152 180	40 68 96 123 151 179
	7	36 55 61 108 132 155 180	33 56 81 107 132 156 181	35 60 85 108 133 157 178	33 59 83 105 133 157 181	33 55 82 106 130 157 180	36 60 84 108 132 157 181

Table 3 The highest objective function values of Kapur's method for the ten test images

Kapur's Method									
Image	K	IEHO	OEHO	DCEHO	EHO	ABC	CS	BAT	PSO
Airplane	5	20.7868	20.7868	20.7868	20.7868	20.7868	20.7868	20.7868	20.6992
	6	23.2116	23.2116	23.2116	23.2116	23.2032	23.2073	23.2111	23.1913
	7	25.4889	25.4836	25.4828	25.4828	25.4773	25.4680	25.4855	25.4810
Baboon	5	20.6171	20.6171	20.6171	20.6171	20.6123	20.6150	20.6136	20.6171
	6	23.1574	23.1574	23.1574	23.1574	23.1414	23.1553	23.1544	23.1315
	7	25.5816	25.5811	25.5808	25.5808	25.5737	25.5736	25.5802	25.5782
Bird	5	20.8648	20.8648	20.8648	20.8648	20.8648	20.8606	20.8629	20.8621
	6	23.4979	23.4979	23.4979	23.4979	23.4979	23.4813	23.4825	23.4939
	7	25.9218	25.9209	25.9206	25.9206	25.9031	25.8838	25.9116	25.9039
Fishing	5	19.8621	19.8621	19.8621	19.8621	19.8621	19.8619	19.8605	19.8621
	6	22.6694	22.6689	22.6676	22.6676	22.6694	22.6675	22.6615	22.6676
	7	25.2951	25.2937	25.2937	25.2937	25.2936	25.2938	25.2938	25.2931
Lake	5	20.9166	20.9092	20.9092	20.9092	20.8915	20.9031	20.9159	20.8661
	6	23.4122	23.4122	23.4122	23.4122	23.4109	23.4046	23.4106	23.4119
	7	25.8589	25.8582	25.8579	25.8579	25.8230	25.8235	25.8510	25.8545
Lena	5	20.6154	20.6154	20.6154	20.6154	20.6154	20.6128	20.6154	20.6142
	6	22.9940	22.9940	22.9940	22.9940	22.9927	22.9831	22.9919	22.9918
	7	25.2015	25.2013	25.2013	25.2013	25.1773	25.1603	25.1984	25.1712
Peppers	5	21.4475	21.4475	21.4475	21.4475	21.4473	21.4435	21.4468	21.4473
	6	23.9143	23.9138	23.9138	23.9138	23.9076	23.8965	23.9111	23.8920
	7	26.3119	26.3113	26.3108	26.3108	26.3010	26.2920	26.2995	26.3105
Snake	5	21.5046	21.5046	21.5046	21.5046	21.5046	21.5034	21.5046	21.5046
	6	24.1183	24.1182	24.1182	24.1182	24.1113	24.1064	24.1149	24.1110
	7	26.5874	26.5865	26.5842	26.5842	26.5736	26.5598	26.5831	26.5754
Starfish	5	21.8775	21.8775	21.8775	21.8775	21.8631	21.8760	21.8775	21.8679
	6	24.4659	24.4659	24.4659	24.4659	24.4659	24.4646	24.4651	24.4626
	7	26.9090	26.9090	26.9080	26.9080	26.9052	26.9038	26.9064	26.9033
Women	5	20.7722	20.7722	20.7722	20.7722	20.7719	20.7702	20.7715	20.7722
	6	23.1691	23.1691	23.1691	23.1691	23.1665	23.1649	23.1676	23.1660
	7	25.4313	25.4308	25.4215	25.4215	25.4109	25.4072	25.4206	25.4021

6.2 Metrics for quality evaluation

Three widely used quality metrics, e.g., peak signal to noise ratio (PSNR), structural similarity index metric (SSIM) [39] and feature similarity index metric (FSIM) [40] are used to quantify the performance of different algorithms. PSNR measures the pixel-to-pixel intensity similarity between the reference and the processed image is obtained by computing the mean square error (MSE) between the original image and segmented image whereas SSIM and FSIM are used to verify the structure and feature similarity of the segmented images in comparison of original images. The definition of the metrics are given in the subsequent parts of this section.

6.2.1 Peak signal to noise ratio (PSNR)

Peak signal to noise ratio in dB is defined as, PSNR measures in decibel (dB) as

$$PSNR = 10Log_{10}\left(\frac{255^2}{MAE}\right) \tag{22}$$

where MSE is mean absolute error, formulated as:

$$MSE = \frac{\sum_i^M \sum_{j=1}^N |I(i,j) - S(i,j)|}{M * N} \tag{23}$$

where $I(i,j)$ and $S(i,j)$ in Eq. (23) denote the original image and segmented images of $size(M * N)$ respectively.Higher value of PSNR implies better performance

6.2.2 Structural similarity index metric (SSIM)

SSIM is largely used to measure the structural similarity between original and segmented image. Highest value of

Table 4 The highest objective function values of Otsu's method for the ten test images

Otsu's Method								
Image	K	IEHO	EHO	ABC	CS	BAT	PSO	DP
Airplane	5	1980.1921	1980.1921	1980.1921	1980.1921	1980.1921	1980.1921	1980.1277
	6	1994.5697	1994.5697	1994.5579	1994.5579	1994.5660	1994.5493	1994.4211
	7	2004.6735	2004.6923	2004.6481	2004.6478	2004.6509	2004.6478	2004.4846
Baboon	5	1616.2309	1616.2309	1616.2309	1616.2309	1616.2309	1616.2309	1616.1575
	6	1632.2973	1632.2959	1632.2480	1632.2641	1632.2025	1632.2725	1632.2025
	7	1643.2574	1643.2854	1643.0157	1643.2199	1643.1284	1643.2222	1643.1223
Bird	5	2764.1810	2764.1810	2764.1810	2764.1810	2764.1810	2764.1810	2764.1404
	6	2776.0910	2776.0883	2776.0591	2776.0363	2776.0259	2776.0625	2776.0161
	7	2784.7716	2784.7671	2784.6361	2784.7158	2784.6513	2784.7671	2784.5354
Fishing	5	5355.8555	5355.8555	5355.8555	5355.8555	5355.8555	5355.8555	5355.8233
	6	5376.4220	5376.4197	5376.3996	5376.4115	5376.4116	5376.4115	5376.3448
	7	5390.2085	5390.2012	5390.1333	5390.1641	5390.1365	5390.1692	5390.1196
Lake	5	3987.9685	3987.9685	3984.2283	3987.8894	3987.8939	3983.9939	3987.7022
	6	4007.5480	4007.5480	4007.5364	4007.4406	4007.4667	4007.5088	4007.3063
	7	4020.1786	4020.1731	4020.1440	4020.1620	4020.1620	4020.1413	4019.5827
Lena	5	2218.8381	2218.8381	2218.8381	2218.8381	2218.8381	2218.8381	2218.7615
	6	2239.4996	2239.4996	2239.4517	2239.4928	2239.4313	2239.4928	2239.3588
	7	2250.3379	2250.3379	2250.0341	2250.2688	2250.142	2250.3019	2250.1420
Peppers	5	3197.0538	3197.0538	3197.0355	3197.0538	3197.0538	3197.0538	3197.0355
	6	3223.6905	3223.6850	3223.6570	3223.6085	3223.5282	3223.6850	3223.5282
	7	3241.3072	3241.2994	3241.0721	3241.1450	3241.0759	3240.8894	3240.1492
Snake	5	1317.0273	1317.0273	1317.0273	1317.0273	1317.0273	1317.0273	1317.0066
	6	1336.1718	1336.1718	1336.0328	1336.0917	1336.1275	1336.1608	1336.0682
	7	1348.9271	1348.9246	1348.8866	1348.9027	1348.8533	1348.9194	1348.4956
Starfish	5	2912.8589	2912.8452	2909.5771	2912.7638	2912.8589	2912.4055	2912.4118
	6	2941.7280	2941.7280	2941.5888	2941.7167	2941.6628	2941.6506	2941.6417
	7	2960.1581	2960.1422	2960.1015	2960.0516	2960.1015	2960.1188	2959.9376
Women	5	1964.8183	1964.8183	1964.6736	1964.8183	1964.8183	1964.7721	1964.7253
	6	1987.7105	1987.7105	1987.3671	1987.5968	1987.6400	1987.7092	1987.0521
	7	2002.5204	2002.5142	2002.4219	2002.4146	2002.3428	2002.3595	2002.1116

SSIM represents the better performance. SSIM is defined as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{24}$$

where μ_x *and* μ_y mean intensity of original image and segmented image σ_x, σ_y are the standard deviation of original and segmented image;σ_{xy} is the covariance of original and segmented image. c_1 *and* c_2 are two constant, such that C_1 =0.01 and C_2=0.03.

6.2.3 Feature similarity index metric (FSIM)

FSIM is, largely used to find the feature similarity between original and segmented image for two images $f_1(X)$ and $f_2(X)$, the FSIM is given by,

$$FSIM = \frac{\sum_{X_\epsilon\Omega} S_L(X)PC_m(X)}{\sum_{X_\epsilon\Omega PC_m(X)}} \tag{25}$$

where Ω means spatial domain of whole image and $S_L(X) = S_{pc}(X)S_G(X)\,S_{pc}(X)S_G(X)$ are given by,

$$\begin{aligned} S_pc(X) &= \frac{2PC_1(X)PC_2(X) + T_1}{PC_1^2(x) + PC_2^2 + T_1} \; and \\ S_G(X) &= \frac{2G_1(X)G_2(X) + T_2}{G_1^2(X) + G_2^(X) + T_2} \end{aligned} \tag{26}$$

where PC_1 and PC_2 are the phase congruency maps take out from two images $f_1(X)$ and $f_2(X)$ respectively; T_1 and T_2 are constants and taken as $T_1 = 0.85, T_2 = 160$. Higher value of FSIM indicates better performance.

Table 5 Comparison between IEHO, EHO, ABC, CS, BAT and, PSO in terms of PSNR, SSIM and FSIM taking Kapur's entropy as objective function for ten test images

Image																		
K	PSNR(dB)						SSIM						FSIM					
	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Airplane																		
5	22.7735	22.7735	22.7735	22.7735	22.7735	22.7735	0.9648	0.9648	0.9648	0.9648	0.9648	0.9648	0.9128	0.9128	0.9128	0.9128	0.9128	0.9128
6	24.2293	24.2293	24.2278	24.2278	24.2293	24.2293	0.9698	0.9698	0.9692	0.9692	0.9696	0.9697	0.9378	0.9378	0.9368	0.9368	0.9372	0.9371
7	25.4619	25.3533	25.3052	25.3006	25.3632	25.3006	0.9754	0.9739	0.9733	0.9739	0.9742	0.9739	0.9552	0.9511	0.9503	0.9501	0.9532	0.9501
Baboon																		
5	24.8027	24.8027	24.8027	24.8027	24.8027	24.8027	0.9917	0.9917	0.9917	0.9917	0.9917	0.9917	0.9653	0.9653	0.9653	0.9653	0.9653	0.9653
6	25.8476	25.764	25.823	25.6815	25.8364	25.7898	0.9932	0.9930	0.9932	0.9928	0.9931	0.9931	0.9745	0.9732	0.9741	0.9733	0.9735	0.9726
7	26.7114	26.7114	26.5594	26.7093	26.6432	26.5517	0.9945	0.9945	0.9939	0.9942	0.9942	0.9935	0.9824	0.9824	0.9811	0.9805	0.9815	0.9797
Bird																		
5	22.4897	22.4897	22.4897	22.4897	22.4897	22.4897	0.8938	0.8938	0.8938	0.8938	0.8938	0.8938	0.8638	0.8638	0.8638	0.8638	0.8638	0.8638
6	26.3513	26.2841	26.332	26.1287	26.3132	26.2682	0.9951	0.9949	0.9951	0.9949	0.9950	0.9949	0.9002	0.8987	0.8992	0.8988	0.8992	0.8989
7	26.1491	25.1503	24.9766	25.0635	25.1432	25.0755	0.9354	0.9285	0.9230	0.9257	0.9334	0.9284	0.8955	0.8866	0.8857	0.8863	0.8925	0.8853
Fising																		
5	18.6896	18.6896	18.6896	18.6896	18.6896	18.6896	0.9064	0.9064	0.9064	0.9064	0.9064	0.9064	0.8328	0.8328	0.8328	0.8328	0.8328	0.8328
6	20.0968	19.7122	19.6873	19.6896	19.8922	19.6896	0.9854	0.9421	0.9399	0.9400	0.9814	0.9400	0.8834	0.868	0.8656	0.8656	0.8754	0.8656
7	20.5931	20.5931	20.4579	20.5684	20.4951	20.5789	0.9540	0.954	0.9555	0.9523	0.9520	0.9524	0.8935	0.8905	0.8917	0.8923	0.8915	0.8898
Lake																		
5	18.9654	18.9654	18.9654	18.9654	18.9654	18.9654	0.7492	0.7492	0.7492	0.7492	0.7492	0.7492	0.9014	0.9014	0.9014	0.9014	0.9014	0.9014
6	24.6196	24.6196	24.5968	24.5736	24.5976	24.5578	0.9918	0.9915	0.9916	0.9916	0.9916	0.9916	0.9538	0.9538	0.9537	0.9538	0.9535	0.9537
7	25.7125	25.7125	25.5973	25.5421	25.6754	25.5893	0.9925	0.9924	0.9920	0.9920	0.9921	0.9921	0.9638	0.9637	0.9628	0.9627	0.9629	0.9628
Lena																		
5	19.0539	19.0539	19.0539	19.0539	19.0539	19.0539	0.7956	0.7956	0.7956	0.7956	0.7956	0.7956	0.8829	0.8829	0.8829	0.8829	0.8829	0.8829
6	20.433	20.4330	20.4308	20.4269	20.4319	20.4269	0.8310	0.8310	0.8309	0.8310	0.8308	0.8310	0.9045	0.9045	0.9044	0.9039	0.9043	0.9039
7	20.8721	20.8721	20.8561	20.7741	20.8654	20.7828	0.8355	0.8355	0.8354	0.8323	0.8333	0.8323	0.9157	0.9157	0.9167	0.9157	0.9154	0.9150
Peppers																		
5	21.6096	21.6096	21.5879	21.6096	21.6079	21.6096	0.8207	0.8207	0.8201	0.8207	0.8205	0.8207	0.8850	0.8849	0.8850	0.8849	0.8849	0.8849
6	23.5938	23.2743	23.2677	23.2693	23.2703	23.2558	0.8759	0.8670	0.864	0.8696	0.8689	0.8670	0.9100	0.9065	0.9065	0.9065	0.9061	0.9064
7	24.7635	24.4348	24.3602	24.4349	24.4612	24.3875	0.8785	0.8759	0.8791	0.8769	0.8772	0.8761	0.9282	0.9258	0.9258	0.9257	0.9267	0.9267
Snake																		
5	20.8403	20.8403	20.8403	20.8403	20.8403	20.8403	0.8625	0.8625	0.8625	0.8625	0.8625	0.8625	0.8595	0.8595	0.8595	0.8595	0.8595	0.8595
6	21.9981	21.9981	21.8609	21.9946	21.9009	21.9933	0.8833	0.8833	0.8791	0.8833	0.8830	0.8833	0.8824	0.8824	0.8791	0.8820	0.8821	0.8823
7	26.4755	26.3845	26.4594	26.4323	26.3735	26.3656	0.9925	0.9922	0.9925	0.9923	0.9921	0.9922	0.9505	0.9492	0.9496	0.9504	0.9495	0.9498

Table 5 (continued)

Image	K	PSNR(dB)						SSIM						FSIM					
		IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Starfish																			
	5	22.5054	22.4423	22.1770	22.4251	22.4251	22.3513	0.9827	0.9823	0.9778	0.9822	0.9823	0.982	0.8386	0.8359	0.8234	0.8358	0.8353	0.8336
	6	23.7429	23.7429	23.6976	23.6811	23.6543	23.7415	0.9861	0.9861	0.9859	0.9857	0.9858	0.986	0.8672	0.8672	0.8663	0.8657	0.8657	0.8652
	7	22.6251	22.6251	22.6222	22.6217	22.6223	22.6182	0.8452	0.8452	0.8451	0.8455	0.8457	0.8455	0.8679	0.8679	0.8674	0.8672	0.8671	0.8681
Women																			
	5	22.6484	22.6484	22.6236	22.6484	22.6342	22.6164	0.8597	0.8597	0.8556	0.8597	0.8577	0.8587	0.7893	0.7893	0.7879	0.7893	0.7893	0.7877
	6	24.0085	24.0085	23.9170	23.9685	23.9431	23.9685	0.8864	0.8864	0.8850	0.8827	0.8843	0.8827	0.8123	0.8123	0.8089	0.8110	0.8116	0.8110
	7	25.8438	25.8338	25.7429	25.7690	25.7154	25.7967	0.9818	0.9817	0.9810	0.9811	0.9815	0.9816	0.8541	0.8531	0.8521	0.8517	0.8519	0.8528

6.3 Simulation results

We have simulated five other popular meta-heuristic algorithms along with the proposed one to compare them with the proposed algorithm. The performance is compared in view of solution quality, the stability of the algorithms, convergence speed, and execution time.

6.3.1 Solution quality

The optimized thresholds value of Kapur's entropy for 5-level, 6-level and 7-levels of all the images are listed in Table 2 and the objective function values of Kapur's entropy and Otsu's between class variance are presented in Tables 3 and 4. In Table 3 we have also compute the individual performace of the OBL and DC with EHO, and shown as OEHO and DCEHO column in the Table 3. These results confirm that the proposed IEHO algorithm gives better objective functions values with respect the other methods. As the individual performance of the OBL and DC with EHO is not better,we did not consider this two technique in others tables. It is also observed that the performance the proposed algorithm is gradually improving with the increase of the number of threshold levels. This means that the IEHO can cover larger domain of the problems with respect to the other algorithms.

Segmented images obtained after 5-level, 6-level and 7-levels segmentation of lake, peppers and women are shown in Figs. 3, 4 and 5. The quality of the segmented images is then evaluated by calculating PSNR, SSIM, and FSIM. The measured values of PSNR, SSIM, and FSIM by the Kapur's entropy and Otsu's between class variance objective function are shown in Tables 5 and 6 respectively. We see from the table that, IEHO shows better performance in terms of PSNR, SSIM, and FSIM value than the conventional EHO and the other algorithms.

6.3.2 Steadiness of the algorithm

Since the optimization problems are random in nature and the initial population is produced through a random search, the value of the optimized objective function may slightly vary in each execution. Hence, the performance of the algorithms must be verified by computing mean and standard deviation of the values of optimized objective functions through several executions of the algorithm. A higher mean value indicates the better accuracy and a lesser value of the standard deviation suggests higher stability of the algorithm. Incorporation of DCM into the standard EHO has elevated the possibility of producing the same result in each run. It increases the mean value, whereas it decreases the value of standard deviation of the results compared the conventional

Table 6 Comparison between IEHO, EHO, ABC, CS, BAT and, PSO in terms of PSNR, SSIM and, FSIM taking Otsu's between-class variance objective function for ten test images

Image																		
K	PSNR(dB)						SSIM						FSIM					
	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Airplane																		
5	21.4818	21.4818	21.4818	21.4818	21.4818	20.9451	0.9792	0.9792	0.9792	0.9792	0.9792	0.9748	0.9168	0.9168	0.9168	0.9168	0.9168	0.9085
6	22.3024	22.3024	22.1060	22.1072	22.1056	21.8522	0.9785	0.9785	0.9777	0.9784	0.9782	0.9785	0.9257	0.9257	0.9255	0.9246	0.9247	0.9216
7	24.8753	24.7551	23.4647	23.8101	24.5876	24.4321	0.9949	0.9948	0.9942	0.9943	0.9941	0.9948	0.9482	0.9456	0.9394	0.9428	0.9462	0.9446
Baboon																		
5	21.4198	21.4198	21.3098	21.2329	21.2329	21.4198	0.9695	0.9695	0.9691	0.9701	0.9695	0.9695	0.9479	0.9479	0.9485	0.9467	0.9479	0.9479
6	22.9111	22.9111	22.8565	22.9081	22.2095	22.2105	0.9808	0.9808	0.9790	0.9805	0.9706	0.9785	0.9614	0.9612	0.9597	0.9612	0.9612	0.9543
7	24.5654	24.5654	24.4304	24.5622	24.4002	24.4193	0.9909	0.9909	0.9905	0.9909	0.9907	0.9906	0.9523	0.9493	0.9513	0.9487	0.9493	0.9473
Bird																		
5	21.6393	21.6393	21.6393	21.6276	21.6393	21.4910	0.8892	0.8892	0.8892	0.8923	0.8892	0.8905	0.8711	0.8711	0.8712	0.8720	0.8711	0.8711
6	22.7619	22.7619	22.7619	22.614	22.6547	22.7271	0.9085	0.9085	0.9082	0.9017	0.9076	0.9084	0.8867	0.8867	0.8857	0.8840	0.8862	0.8860
7	25.5801	25.5332	25.1476	24.3976	24.4365	24.5397	0.9940	0.9940	0.9939	0.9932	0.9939	0.9937	0.9074	0.9045	0.9015	0.9049	0.9059	0.9068
Fishing																		
5	19.9332	19.9332	19.9332	19.9231	19.9332	19.9332	0.9231	0.9231	0.9231	0.9225	0.9231	0.9231	0.8135	0.8135	0.8135	0.8122	0.8135	0.8135
6	22.1056	21.9287	21.9287	21.7692	21.7939	21.864	0.9570	0.9551	0.9541	0.9535	0.9549	0.9540	0.8459	0.8450	0.8450	0.8444	0.8453	0.8445
7	22.9925	22.9925	22.7593	22.7059	22.9543	22.9458	0.9618	0.9618	0.9616	0.9581	0.9617	0.9617	0.8728	0.8728	0.8713	0.8648	0.8724	0.8719
Lake																		
5	24.4136	24.2129	24.0707	24.242	24.2976	24.3666	0.9931	0.9925	0.9927	0.9916	0.9929	0.992	0.9488	0.9474	0.9482	0.9437	0.9480	0.9417
6	22.5459	22.5459	22.4904	22.3339	22.5342	22.5411	0.9269	0.9269	0.9178	0.9096	0.9267	0.9266	0.9413	0.9413	0.9394	0.9362	0.9411	0.9412
7	23.1578	23.1578	22.9132	23.0877	23.1274	23.104	0.9618	0.9618	0.9594	0.9597	0.9602	0.9552	0.9512	0.9512	0.9474	0.9501	0.9508	0.9487
Lena																		
5	19.5607	19.5607	19.5607	19.4874	19.5607	19.5500	0.8262	0.8262	0.8262	0.823	0.8262	0.8261	0.8735	0.8735	0.8735	0.8731	0.8735	0.8733
6	20.5425	20.5425	20.5381	20.4358	20.5324	20.5203	0.8371	0.8371	0.8371	0.834	0.8367	0.8370	0.9031	0.9031	0.9016	0.9006	0.9019	0.9008
7	20.9234	20.8771	20.5306	20.6354	20.7658	20.6260	0.8418	0.8418	0.8283	0.8318	0.8396	0.8376	0.9150	0.9143	0.9112	0.9103	0.9133	0.9068
Peppers																		
5	21.7189	21.7189	21.6929	21.7131	21.6929	21.6929	0.8210	0.821	0.8207	0.8204	0.8207	0.8207	0.8860	0.886	0.8851	0.8847	0.8851	0.8851
6	22.1698	22.1000	22.0781	22.0486	22.0653	21.8492	0.8190	0.820	0.8207	0.8183	0.8185	0.8174	0.9183	0.8982	0.898	0.8971	0.9142	0.9022
7	23.4798	23.4798	23.3313	23.312	23.4751	23.4742	0.8864	0.8864	0.8780	0.8818	0.8846	0.8782	0.9220	0.922	0.9196	0.9209	0.9220	0.9218
Snake																		
5	19.2832	19.2832	19.2832	19.2016	19.2832	19.2832	0.8420	0.8420	0.8420	0.8406	0.8420	0.8420	0.8420	0.8420	0.8420	0.8418	0.8420	0.8420
6	21.0233	21.0233	20.8378	20.6482	20.8769	20.6994	0.8904	0.8904	0.8889	0.8776	0.8871	0.8774	0.8824	0.8824	0.8811	0.8742	0.8820	0.877
7	23.7446	22.1364	22.1364	22.086	22.3586	21.9332	0.9199	0.9160	0.9131	0.9034	0.9139	0.9055	0.9241	0.9045	0.9029	0.9034	0.9038	0.8985

Table 6 (continued)

Image K	PSNR(dB)						SSIM						FSIM					
	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Starfish																		
5	19.8976	19.8976	19.8976	19.8425	19.8976	19.8976	0.7850	0.7850	0.7850	0.7837	0.7850	0.7850	0.7879	0.7879	0.7879	0.7867	0.7879	0.7879
6	21.1973	21.1973	21.1901	21.1414	21.1424	21.1354	0.8187	0.8187	0.8187	0.8147	0.8182	0.8180	0.8254	0.8254	0.8249	0.8254	0.8247	0.8244
7	22.0896	22.0896	21.9830	22.0230	22.0654	22.0591	0.8386	0.8386	0.833	0.8371	0.8375	0.8370	0.8526	0.8526	0.8496	0.8504	0.8517	0.8505
Women																		
5	22.1814	22.1814	22.1771	22.1369	22.1814	22.1814	0.8375	0.8375	0.8369	0.8372	0.8375	0.8375	0.7711	0.7711	0.7710	0.7694	0.7711	0.7711
6	23.2129	23.2129	23.1693	23.1759	23.1974	23.1317	0.8566	0.8566	0.8562	0.8569	0.8563	0.8561	0.7952	0.7952	0.7948	0.7941	0.7938	0.7931
7	24.8235	24.5390	24.4078	24.5132	24.6754	24.4887	0.8883	0.8883	0.8791	0.8865	0.8882	0.8881	0.8195	0.8193	0.8191	0.8183	0.8172	0.8160

EHO. Tables 7 and 8 display the values of mean and standard deviation for two optimized objective functions by executing each algorithm 30 times (i.e.30 runs).For example, in case of airplane image with 7-level,the mean of objective function values are 25.4804, 25.4739, 25.433, 25.3866, and 25.4413 and standard deviations are 0.0007474, 0.0087591, 0.0460967, 0.0587565, and 0.010462 respectively. Hence, the proposed IEHO algorithm performs better with respect to the others in segmentation with higher number of threshold levels.

6.3.3 Convergence and computational Time

The convergence graph for different algorithms using two objective functions are shown in Fig. 6 for lena and fishing images. From this figure we find that the convergence rate of conventional EHO has dramatically increased after incorporating the OBL and DC into the conventional EHO. The improved EHO has been converged within 20 to 30 iterations whereas average convergence rate of standard EHO together with other algorithms is above 50 iterations.

The average execution time is measured to compare the computational complexity of the different algorithms used for multilevel thresholding. The average execution time of IEHO, EHO, ABC, CS, BAT and, PSO are shown in Table 9. Each of the algorithms is executed 30 times to calculate the average execution time. It is evident from the table that IEHO base segmentation is faster than other.

7 Conclusions

This paper proposes an OBL based improved elephant herding algorithm incorporating dynamic Cauchy mutation to enrich the performance of the standard EHO. OBL is employed to accelerate the conventional EHO, and DCM is incorporated to reduce the possibility of being confined in local optima. The proposed IEHO is applied to multilevel thresholding for image segmentation. The images under test are taken from standard Berkeley Segmentation Dataset. The performance of IEHO is compared with some recently proposed evolutionary and classical optimization algorithms. From the experimental results, we notice that the proposed IEHO outperforms the conventional EHO, ABC, CS, PSO and DP both in terms of quality at higher level thresholding and convergence rate. This suggests that we can use the proposed algorithm effectively for multilevel thresholding in image segmentation for different applications.

Table 7 List of mean and standard deviation after 30 runs for each algorithm calculated on ten images taking Kapur's objective function

Image	K	MEAN						SD					
		IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Airplane	5	20.7858	20.7841	20.7824	20.7751	20.7450	20.7751	0.0003604	0.0143759	0.0044567	0.0197168	0.0446489	0.0032176
	6	23.2112	23.2110	23.1462	23.1449	23.1740	23.1442	0.0009520	0.0010744	0.063528	0.0299690	0.0464008	0.0456581
	7	25.4804	25.4739	25.433	25.3866	25.4552	25.4413	0.0007474	0.0087591	0.0460967	0.0587565	0.0281845	0.0104620
Baboon	5	20.6170	20.6169	20.5833	20.5865	20.5431	20.5848	0.0002544	0.0005432	0.0284150	0.0228524	0.0601027	0.0025011
	6	23.1572	23.1572	23.0876	23.0929	23.1090	23.1480	0.0003922	0.0005511	0.0368363	0.0415197	0.0577292	0.0233072
	7	25.5762	25.5663	25.4572	25.4562	25.4761	25.5222	1.08E-14	0.0035185	0.1113593	0.0614911	0.1239543	0.1223237
Bird	5	20.8604	20.8599	20.8627	20.8466	20.7519	20.7813	0.0001143	0.0045723	0.0088729	0.0093853	0.2256009	0.0992359
	6	23.4965	23.4962	23.4828	23.4080	23.3457	23.4014	0.0022307	0.0024628	0.0135114	0.0546750	0.2257058	0.1533125
	7	25.8751	25.8755	25.7727	25.8114	25.7951	25.8182	0.01573570	0.0166281	0.2341654	0.0501525	0.0988382	0.1545074
Fishing	5	19.8620	19.8618	19.7704	19.8519	19.7974	19.8290	0.0002614	0.0003567	0.0934285	0.0091038	0.1002568	0.1248983
	6	22.6674	22.6671	22.6640	22.6322	22.5980	22.6510	0.0042328	0.0058095	0.008311	0.0264873	0.0510764	0.0376507
	7	25.2894	25.2814	25.1410	25.2328	25.2155	25.2561	0.0060530	0.0074511	0.1994503	0.0240823	0.0570252	0.0864679
Lake	5	20.9156	20.9155	20.8373	20.8872	20.8931	20.6138	0.0014884	0.0015444	0.0284286	0.0113810	0.0511355	0.1348177
	6	23.4108	23.4106	23.4005	23.3704	23.3872	23.3545	0.0035467	0.0045096	0.0077408	0.0308819	0.0360644	0.1157483
	7	25.8342	25.8162	25.7533	25.7436	25.7603	25.7490	0.0278717	0.0303790	0.0609403	0.0497363	0.0612546	0.1489812
Lena	5	20.6144	20.6131	20.5057	20.6026	20.6039	20.6121	7.23E-10	0.0004010	0.1439334	0.0023791	0.0283830	0.0088027
	6	22.9840	22.9760	22.9716	22.9433	22.9627	22.8875	7.23E-12	0.0033417	0.0181511	0.0306668	0.0385689	0.1520024
	7	25.1867	25.1712	25.1547	25.106	25.1653	25.0547	0	7.23E-15	0.1932272	0.0246326	0.0215277	0.0353236
Peppers	5	21.4452	21.4448	21.4376	21.4312	21.4209	21.3848	0	0.0004726	0.0093771	0.0124795	0.0897395	0.0855788
	6	23.9085	23.9028	23.812	23.8662	23.8717	23.8729	0	0.0045983	0.0084492	0.0244000	0.0347036	0.0908805
	7	26.2876	26.2658	26.2353	26.2551	26.2555	26.1887	0.0101513	0.0139308	0.0303879	0.0312554	0.0362159	0.1850169
Snake	5	21.5046	21.5046	21.5032	21.4976	21.4895	21.4229	4.51E-05	5.38E-05	0.0012139	0.0049375	0.0145249	0.0157192
	6	24.1180	24.1178	24.1099	24.0869	24.0972	24.1177	0.0003225	0.0013004	0.0052714	0.0144312	0.0152422	0.0013339
	7	26.5804	26.5774	26.5556	26.5202	26.5462	26.4406	0.0003963	0.0006914	0.21034	0.0148209	0.0259040	0.0253465
Starfish	5	21.8771	21.8769	21.8267	21.8619	21.8644	21.5987	0	0.000112	0.0303219	0.0117868	0.0323512	0.2541156
	6	24.4609	24.4575	24.4263	24.4384	24.4455	24.3079	0.0069068	0.0099965	0.0709396	0.0188376	0.0220129	0.0298974
	7	26.9070	26.9052	26.8913	26.8596	26.7943	26.8335	1.08E-14	0.0007058	0.0134228	0.0292385	0.2226104	0.1022946
Women	5	20.7716	20.7716	20.7586	20.7583	20.7520	20.7245	0	0.0001887	0.0126876	0.0099244	0.0555884	0.0913339
	6	23.1456	23.1249	23.1407	23.1060	23.1535	23.0490	1.81E-14	0.0070681	0.0301770	0.0420412	0.0135006	0.1496251
	7	25.3615	25.3415	25.3401	25.2297	25.3309	25.2408	0.0336280	0.0425729	0.2151559	0.0445864	0.1451749	0.0434630

Table 8 List of mean and standard deviation after 30 runs for each algorithm calculated on ten images taking Otsu's between-class variance objective function

Image	K	MEAN						SD					
		IEHO	EHO	ABC	CS	BAT	PSO	IEHO	EHO	ABC	CS	BAT	PSO
Women	5	1980.1921	1980.1921	1980.1537	1980.1618	1979.5913	1980.1885	0	9.25E-13	0.0360910	0.0442536	0.5941350	0.0131867
	6	1994.4694	1994.4463	1994.3711	1994.3515	1993.1321	1994.3765	0.1193899	0.1234247	0.1809481	0.1651956	1.074033	0.3604835
	7	2004.6542	2004.6065	2004.0903	2004.5301	2004.1857	2004.1364	0.0530161	0.1777638	0.5080393	0.095984	0.5051400	0.8392803
Baboon	5	1616.2004	1616.1936	1616.0477	1616.0103	1613.2932	1616.1077	0.0654833	0.0697458	0.208942	0.2622215	6.8536950	0.3020015
	6	1632.1342	1631.9395	1631.5491	1631.9545	1630.2383	1631.7652	0.256977	0.2727174	0.7990245	0.3047277	1.826669	1.3927723
	7	1642.4321	1641.7671	1642.1670	1642.6156	1641.3840	1642.5242	0.4189119	0.4566921	1.0580275	0.7585746	1.000609	0.857561
Bird	5	2764.1402	2764.1020	2764.0653	2763.9849	2762.4407	2763.0311	0.1251776	0.1538075	0.1653461	0.3733378	3.848186	1.424769
	6	2776.0400	2775.9897	2775.8766	2775.8113	2775.3364	2775.6778	0.0432089	0.0725656	0.1427879	0.2482354	0.870081	1.0195175
	7	2784.7524	2784.7126	2783.8580	2784.1662	2783.4731	2784.1341	0.0481135	0.0639645	0.7244269	0.4610375	1.316247	1.1112881
Fishing	5	5355.6552	5376.4053	5375.9128	5375.8681	5355.6457	5376.1889	0	0.0009021	0.1114314	0.1964761	0.185502	0.014822
	6	5376.4140	5376.4053	5375.9128	5375.8681	5375.6565	5376.1889	0.0221742	0.0230708	0.511592	0.4190524	0.711029	0.6315807
	7	5390.2002	5390.1540	5389.188	5389.5782	5389.3485	5389.8407	0.0377775	0.0448895	0.5465411	0.3085973	0.859776	0.5248261
Lake	5	3987.9672	3987.9667	3981.8143	3987.9452	3982.2818	3987.4166	0.0012774	0.0051572	1.6074128	0.0337085	1.228230	0.8322602
	6	4007.5385	4007.5377	4006.9844	4007.1842	4006.5808	4007.0479	0.0185653	0.020259	0.6269778	0.4030366	0.802618	1.3910382
	7	4020.0106	4019.8361	4019.4153	4019.6027	4019.0306	4019.6882	0.2821623	0.3693839	0.4397651	0.3911579	0.584553	0.4300293
Lena	5	2218.7432	2218.6950	2218.6693	2218.3586	2218.0355	2218.2423	0.0939229	0.1205759	0.6899188	0.1835804	0.657556	0.7119933
	6	2239.4952	2239.4875	2238.7454	2239.1632	2239.0406	2239.1598	0.0109313	0.0412529	0.9696599	0.4055332	0.447194	1.5454972
	7	2250.137	2249.9992	2248.4170	2249.7991	2249.1702	2249.1358	0.2917243	0.3698655	1.0090157	0.5521751	0.8368200	1.21451
Peppers	5	3197.0538	3197.0538	3196.8791	3196.9557	3196.7074	3197.0435	0	1.39E-12	0.2076603	0.1008337	0.392463	0.0167596
	6	3223.6825	3223.6809	3221.9405	3223.1206	3222.8917	3222.9862	0.0144676	0.0150111	1.5115776	0.4842787	0.520670	1.7825516
	7	3241.3002	3241.2817	3238.6139	3240.4438	3240.2080	3240.6811	0.0333134	0.0343068	2.3839984	0.5935454	0.805521	1.3097525
Snake	5	1317.0266	1317.0256	1316.5533	1316.9157	1315.0552	1316.9119	0	0.0092534	0.4421727	0.0947055	1.769889	0.2928141
	6	1336.1541	1336.0726	1335.4361	1335.7888	1335.5354	1336.0026	0.1041717	0.1206239	0.5772474	0.3535092	0.733894	0.2877013
	7	1348.8486	1348.6251	1346.6203	1348.0324	1347.3452	1348.1494	0.1809899	0.2976174	1.4213068	1.4955487	1.267526	0.9501763
Starfish	5	2912.8576	2912.8552	2912.6929	2912.7516	2912.5138	2912.7516	0	0.0061305	2.2310133	1.0912454	0.302809	2.7784642
	6	2941.7134	2941.7097	2941.4055	2941.5575	2940.8394	2941.5575	0.00016549	0.0246332	1.0609124	0.5468551	0.961748	0.2624354
	7	2960.1278	2960.1034	2959.4800	2959.3037	2957.6761	2959.3037	0.00076064	0.0306964	1.6792125	0.5402272	1.267369	1.9932697
Women	5	2912.857	2912.8552	2912.4143	2912.6929	1964.5533	2912.7516	0.0055453	0.0061306	2.2310134	1.0912455	0.218057	2.7784642
	6	2941.7124	2941.7091	2940.7571	2941.4055	1986.9760	2941.5575	0.0239257	0.0248215	1.0609125	0.5468551	0.765256	0.2624354
	7	2002.5004	2002.4891	1999.0212	2001.6962	2001.7380	2002.3247	0.0241811	0.0368406	2.6436939	0.5001488	0.523440	0.2757612

Table 9 Average executing time in seconds of IEHO, EHO, ABC, CS, BAT, and, PSO

Image	K	IEHO	EHO	ABC	CS	BAT	PSO
Airplane	5	95.4	98.5	99.2	96.4	100.1	102.5
	6	118.2	122.3	129.7	121.3	120.4	123.3
	7	124.3	127.8	132.3	126.4	126.9	128.6
Baboon	5	75.6	78.3	80.6	77.4	82.4	79.8
	6	100.1	102.6	104.7	101.5	106.3	104.8
	7	111.5	115.7	117.5	113.6	119.4	117.6
Bird	5	60.7	65.7	67.8	63.2	69.7	68.9
	6	85.3	87.8	89.5	86.7	89.7	89.6
	7	102.5	106.7	105.6	104.8	105.1	107.6
Fishing	5	44.6	46.8	48.7	45.3	49.4	48.9
	6	62.6	64.8	66.7	63.2	65.3	68.9
	7	76.1	78.8	80.2	76.6	77.3	80.1
Lake	5	131.1	137.2	138.9	132.1	135.3	139.8
	6	152.3	154.8	156.8	153.8	155.4	155.8
	7	164.2	166.9	167.8	165.3	171.3	170.8
Lena	5	34.2	37.9	38.5	36.8	37.9	39.7
	6	40.1	43.7	46.7	42.1	44.6	45.7
	7	49.8	53.7	56.8	51.3	55.2	56.8
Peppers	5	31.6	35.7	37.6	33.5	36.5	38.6
	6	53.2	56.7	57.8	55.7	56.4	58.6
	7	85.3	89.6	90.5	87.4	89.3	90.3
snake	5	41.5	42.5	44.6	42.5	41.9	43.7
	6	53.5	57.6	62.1	56.8	57.5	58.5
	7	100.9	103.8	104.3	102.7	104.7	105.7
Starfish	5	125.5	131.6	132.2	128.8	136.7	137.2
	6	143.5	145.5	147.6	144.9	145.8	147.7
	7	163.7	165.7	166.3	164.7	167.9	168.5
women	5	25.2	28.6	30.4	27.4	27.8	30.4
	6	43.7	45.7	46.7	45.1	46.9	47.8
	7	65.7	67.8	70.2	66.7	67.5	70.2

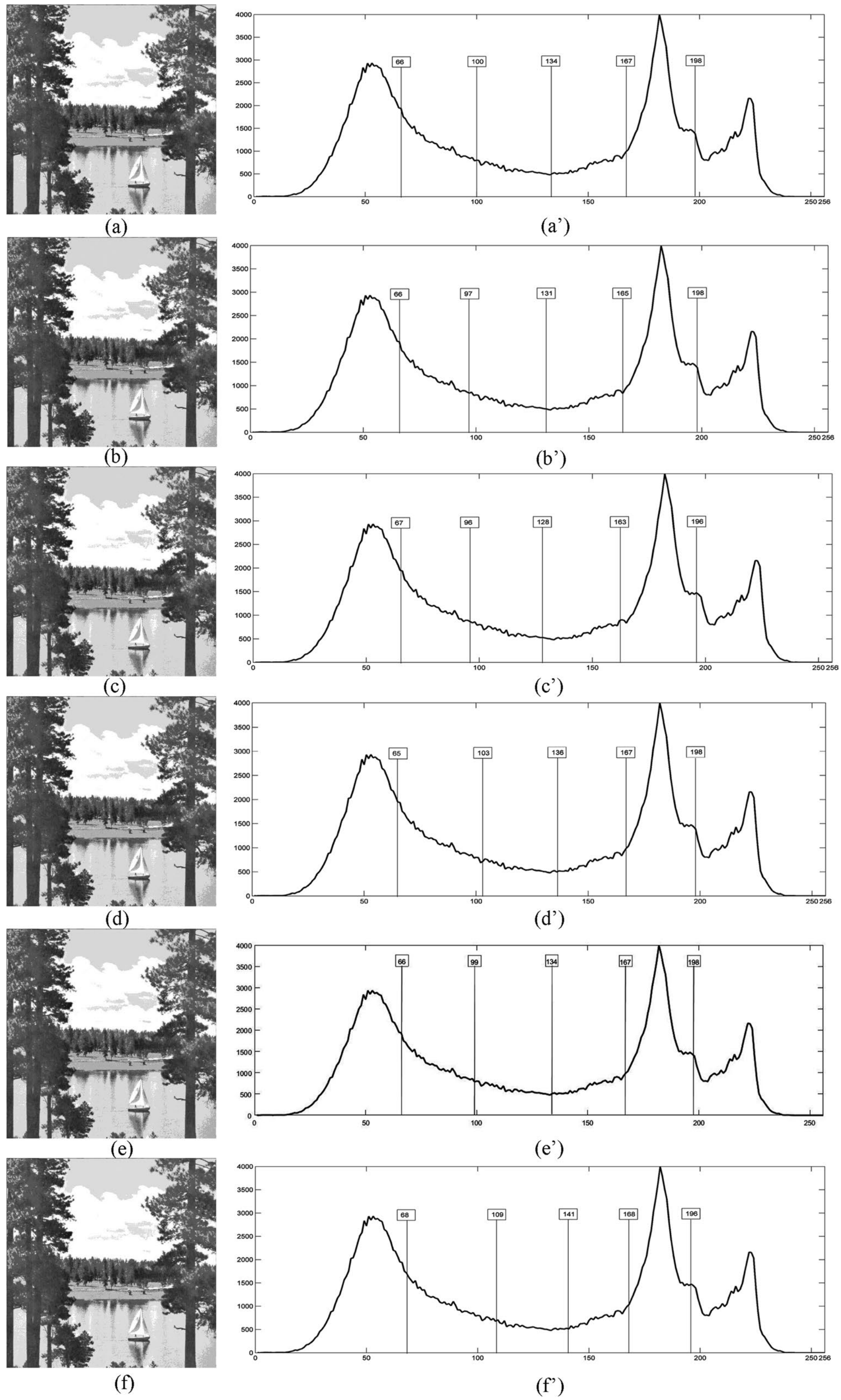


Fig. 3 Images of **a**–**f** are the 5-level thresholded lake images using IEHO, EHO, ABC, CS, BAT, and, PSO taking Kapur's entropy as objective function. (**a'**–**f'**) represents the histogram with the threshold values of lake images respectively

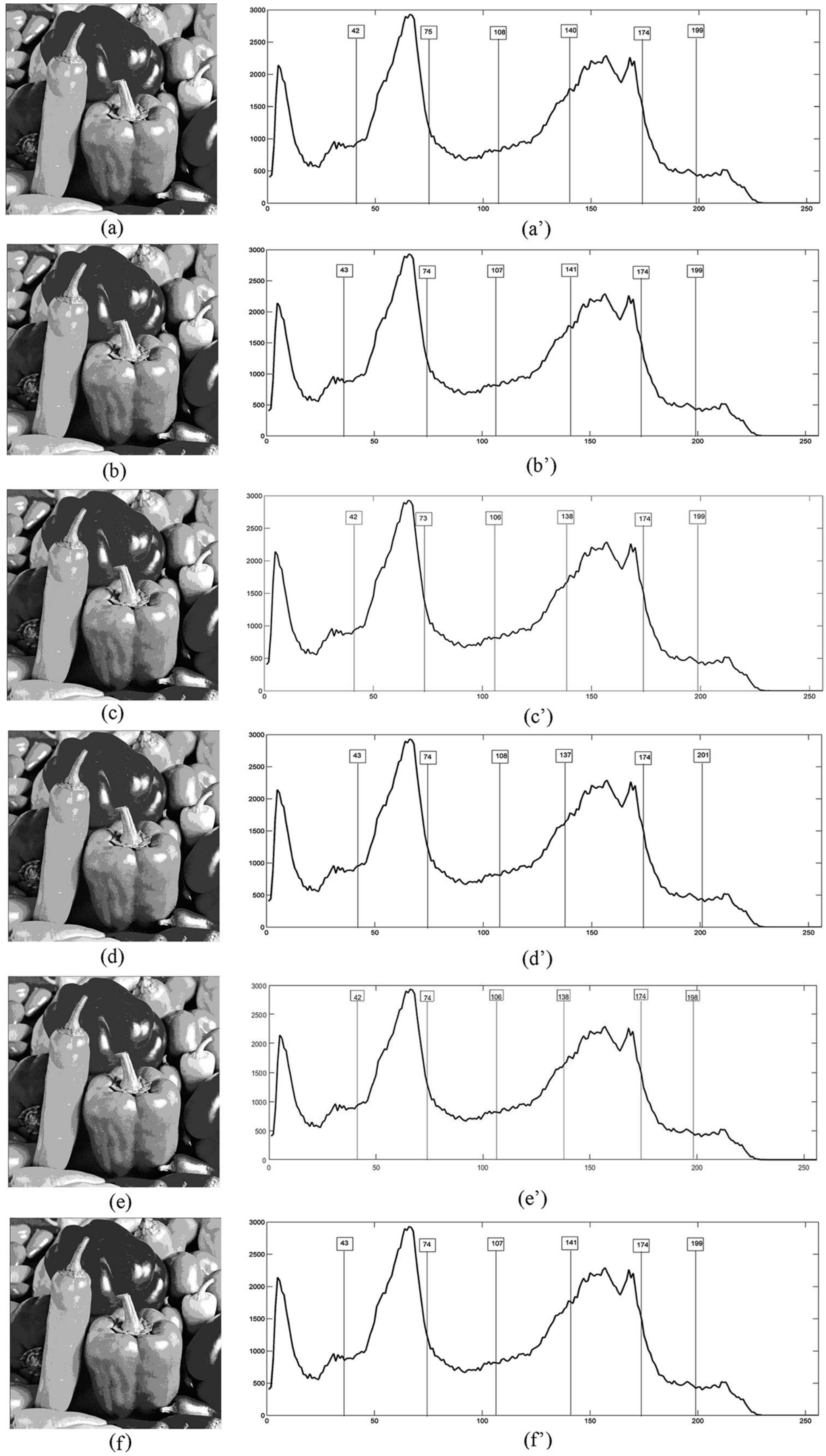
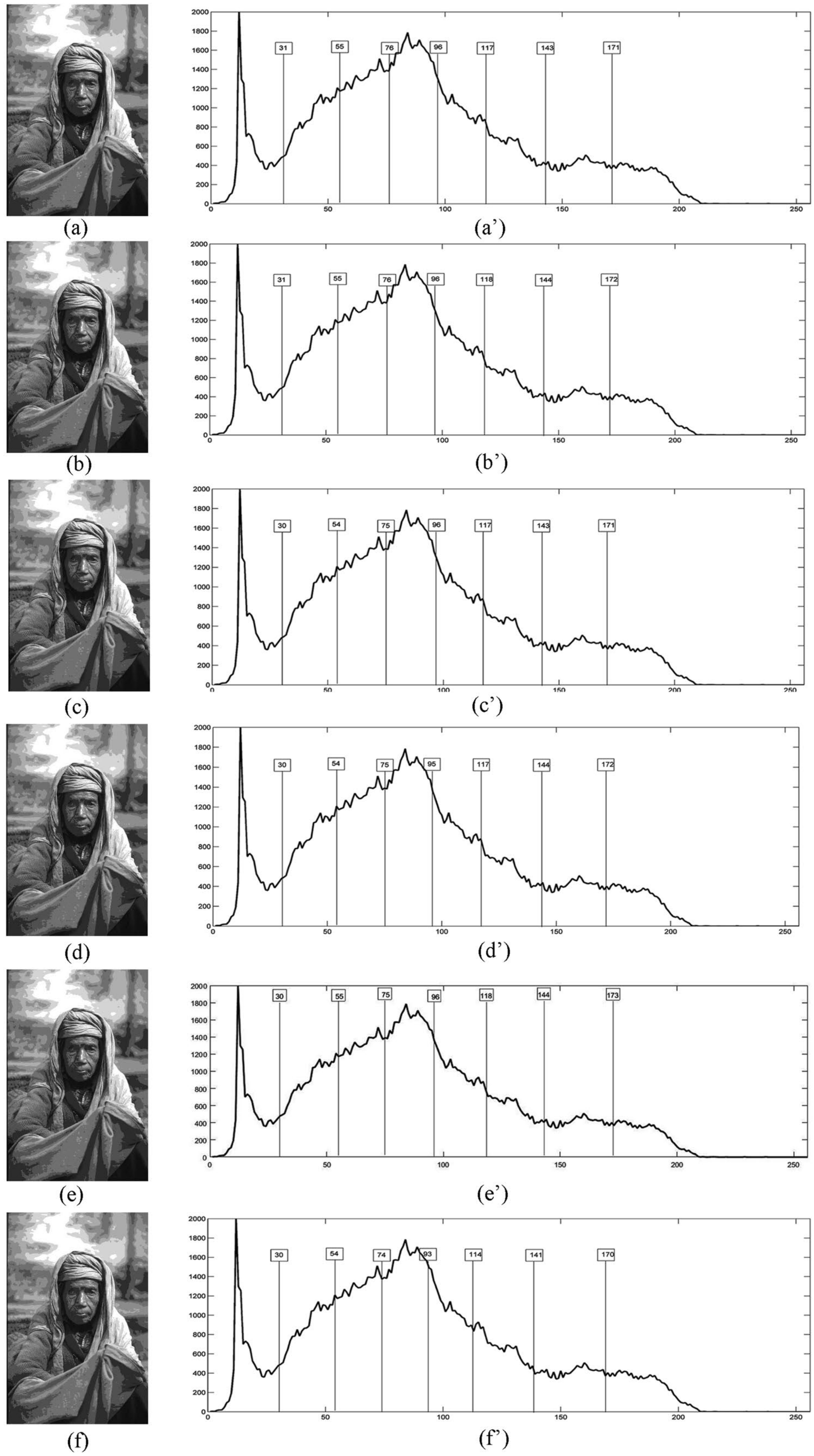


Fig. 4 Images of **a**–**f** are the 6-level thresholded pepper images using IEHO, EHO, ABC, CS, BAT and, PSO taking Kapur's entropy as objective function.(**a'**–**f'**) represents the histogram with the threshold values of pepper images respectively

Fig. 5 Images of **a**–**f** are the 7-level thresholded women images using IEHO, EHO, ABC, CS, BAT, and, PSO taking Otsu's between-class variance objective function.**a'**–**f'** represents the histogram with the threshold values of women images respectively



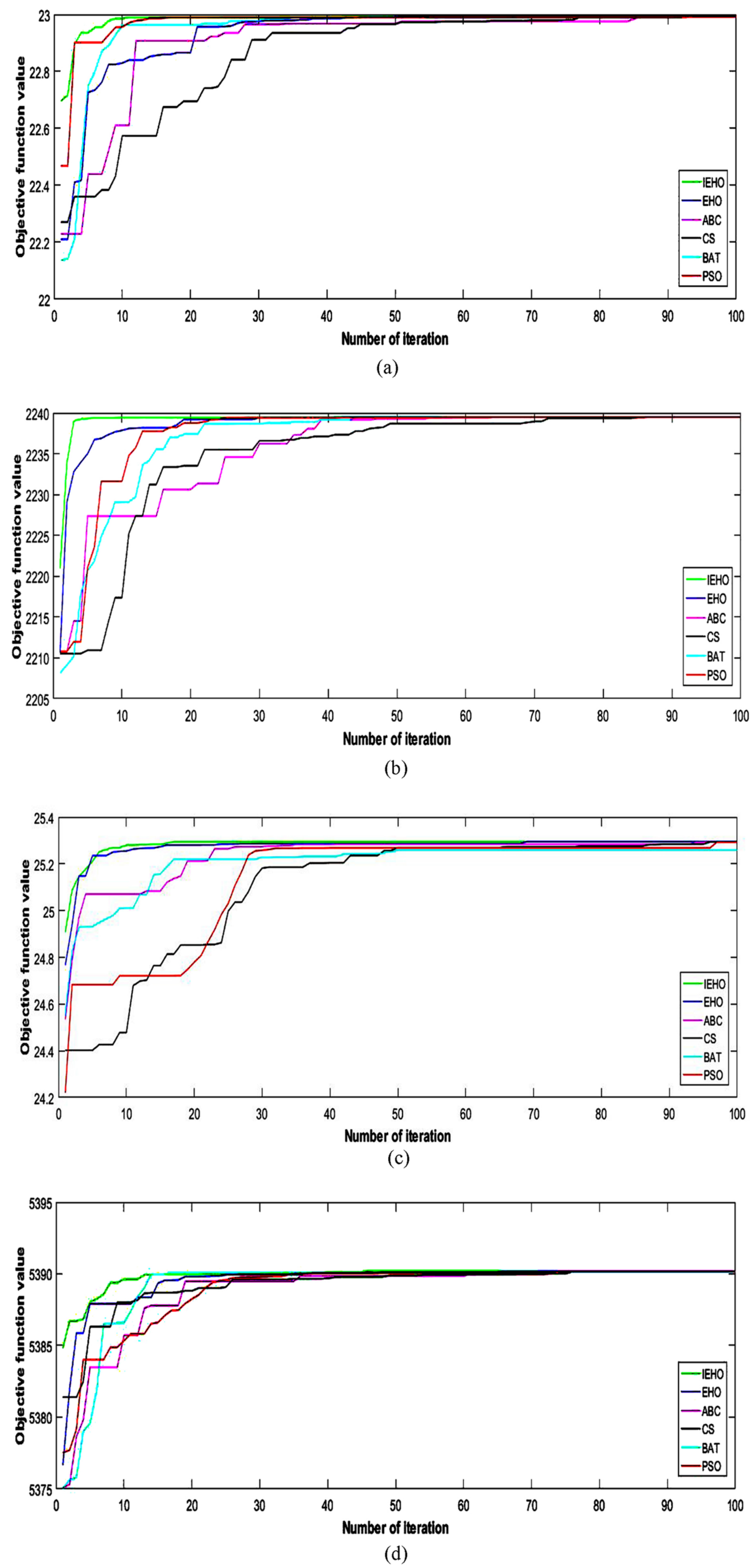


Fig. 6 **a** 6-level convergence graph of lena image by kapur's objective function. **b** 6-level convergence graph of lena image by Otsu's objective function. **c** 7-level convergence graph of fishing image by kapur's objective function. **d** 7-level convergence graph of fishing image by Otsu's objective function

References

1. Wehrens R, Buydens LMC, Lin Z, Lei Z, Xuanqin M (2000) Classical and nonclassical optimization methods. Encycl Anal Chem 1:9678–9689
2. Merzban MH, Elbayoumi M (2019) Efficient solution of otsu multilevel image thresholding: a comparative study. Expert Syst Appl 116:299–309
3. Mousavirad Seyed Jalaleddin, Ebrahimpour-Komleh Hossein (2017) Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms. J Evol Intell 10(1):45–75
4. Mahesh KM, Renjit A (2018) Evolutionary intelligence for brain tumor recognition from MRI images: a critical study and review. J Evol Intell 11(1–2):19–30
5. Yin PY (1999) A fast scheme for multilevel thresholding using genetic algorithms. Signal Processing 72:85–95
6. Hammouche K, Diaf M, Siarry P (2008) A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Comput Vis Image Underst 109(2):163–175
7. Zhang J, Li H, Tang Z, Lu Q, Zheng X, Zhou J (2014) An improved quantum inspired genetic algorithm for image multilevel thresholding segmentation. Math Problems Eng 112:1–12
8. Simon D (2008) Biogeography based optimization. IEEE Trans Evol Comput 12(6):702–713
9. Yin PY (2007) Multilevel minimum cross entropy threshold selection based on particle swarm optimization algorithm. Appl Math Comput 184(2):503–513
10. Kennedy J, Eberhart RC (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw 4:1942–1948
11. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13:3066–3091
12. Bhandari AK, Kumar A, Singh GK (2015) Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's Otsu and Tsallis functions. Expert Syst Appl 42:1573–1601
13. Gandomi AH, Yang XS, Talatahari S, Deb S (2012) Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. Comput Math Appl 63(1):191–200
14. Sathya PD, Kayalvizhi R (2010) Optimum multilevel image thresholding based on Tsallis entropy method with bacterial foraging algorithm. Int J Comput Sci 7(5):336–343
15. Tao W, Jin H, Liu L (2007) Object segmentation using ant colony optimization algorithm and fuzzy entropy. Pattern Recognit Lett 28(7):788–796
16. Agrawal S, Panda R, Bhuyan S, Panigrahi BK (2013) Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol Comput 11:16–30
17. Horng MH (2010) Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst Appl 37:4580–4592
18. Ouadfel S, Taleb-Ahmed A (2016) Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study. Expert Syst Appl 55:566–584
19. Yang XS (2010) A new metaheuristic bat-inspired Algorithm. Stud Comput Intell 284:65–74
20. Bakhshali MA, Shamsi M (2014) Segmentation of color lip images by optimal thresholding using bacterial foraging optimization (BFO). J Comput Sci 5(2):251–257
21. Abdul Kayom M, Khairuzzaman SC (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst Appl 86:64–76
22. El Aziz MA, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256
23. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82
24. Wang H, Wu Z, Rahnamayan S, Liu Y, Ventresca M (2011) Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci 181:4699–4714
25. Yang X, Huang Z (2012) Opposition-based artificial bee colony with dynamic cauchy mutation for function optimization. Int J Adv Comput Technol 4(4):56–62
26. Wang GG, Deb S, Gandomi AH, Alavi AH (2016) Opposition-based kill herd algorithm with Cauchy mutation and position clamping. Neurocomputing 177:147–157
27. Rahnamayan S, Tizhoosh Hamid R, Salama MMA (2008) Opposition-based differential evolution. IEEE Trans Evol Comput 12(1):64–79
28. Wang GG, Deb S, Geo X-Z, Coelho LDS (2016) A new metaheuristic optimization algorithm motivated by elephant herding behavior. Int J Bio-Inspir Comput 8(6):394–409
29. Meena NK, Parashar S, Swarnkar A, Gupta N, Niazi KR (2017) Improved elephant herding optimization for multiobjective DER accommodation in distribution systems. IEEE Trans Ind Inf 14(3):1029–1039
30. Tuba E, Ribic I, Hrosik RC, Tuba M (2017) Support vector machine optimized by elephant herding algorithm for erythemato squamous diseases detection. Information Technology and Quantitative Management (ITQM). Proc Comput Sci 122:916–923
31. Tizhoosh HR (2006) Opposition-based reinforcement learning. J Adv Comput Intell Intell Inf 10(4):578–585
32. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3:82–102
33. Wang H, Liu Y, Zeng SY, Li H, Li C (2007) Opposition-based particle swarm algorithm with Cauchy mutation. In: Proceedings of IEEE congress on evolutionary computation, pp 4750–4756
34. Wang H, Liu Y, Li C, Zeng S (2007) A hybrid particle swarm algorithm with Cauchy mutation. In: IEEE swarm intelligence symposium, Honolulu, Hawaii, pp 356–360
35. Rahnamayan S, Tizhoosh HR, Salama M (2008) Opposition versus randomness in soft computing techniques. Appl Soft Comput 8(2):906–918
36. Rahnamayan S, Tizhoosh Hamid R, Salama MMA (2007) Opposition-based differential evolution(ODE) with variable jumping rate. In: IEEE symposium on foundations of computational intelligence
37. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans SMC 9(1):62–66
38. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Gr Image Process 29:273–285
39. Zhou W, Alan CB, Hamid SRSR, Eero SP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
40. Lin Z, Lei Z, Xuanqin M, Zhang D (2011) FSIM: a feature similarity index for image quality assessment. IEEE Trans Image Process 20(8):2378–2386