Evol. Intel. (2017) 10:45–75
DOI 10.1007/s12065-017-0152-y

RESEARCH PAPER

Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms

Seyed Jalaleddin Mousavirad[1] · Hossein Ebrahimpour-Komleh[1]

Received: 30 December 2016 / Revised: 25 April 2017 / Accepted: 30 May 2017 / Published online: 23 June 2017

Abstract Multilevel thresholding is one of the most broadly used approaches to image segmentation. However, the traditional techniques of multilevel thresholding are time-consuming, especially when the number of the threshold values is high. Thus, population-based metaheuristic (P-metaheuristic) algorithms can be used to overcome this limitation. P-metaheuristic algorithms are a type of optimization algorithms, which improve a set of solutions using an iterative process. For this purpose, image thresholding problem should be seen as an optimization problem. This paper proposes multilevel image thresholding for image segmentation using several recently presented P-metaheuristic algorithms, including whale optimization algorithm, grey wolf optimizer, cuckoo optimization algorithm, biogeography-based optimization, teaching–learning-based optimization, gravitational search algorithm, imperialist competitive algorithm, and cuckoo search. Kapur's entropy is used as the objective function. To conduct a more comprehensive comparison, the mentioned P-metaheuristic algorithms were compared with five others. Several experiments were conducted on 12 benchmark images to compare the algorithms regarding objective function value, peak signal to noise ratio (PSNR), feature similarity index (FSIM), structural similarity index (SSIM), and stability. In addition, Friedman test and Wilcoxon signed rank test were carried out as the nonparametric statistical methods to compare P-metaheuristic algorithms. Eventually, to create a more reliable result, another objective function was evaluated based on Cross Entropy.

Keywords Kapur's entropy · Cross entropy · Metaheuristic algorithms · Image thresholding · Statistical test

✉ Seyed Jalaleddin Mousavirad
jalalmoosavirad@gmail.com;
mousavirad@grad.kashanu.ac.ir

✉ Hossein Ebrahimpour-Komleh
ebrahimpour@kashanu.ac.ir

[1] Department of Computer and Electrical Engineering, University of Kashan, Kashan, Iran

1 Introduction

Image segmentation is one of the most important steps in images analysis. It is the process of dividing an image into some meaningful regions. Pixels inside the same region have similar properties. Image segmentation can be regarded as a preprocessing step in many image processing and computer-vision-based applications [1–4]. In recent years, various image segmentation methods have been proposed such as fuzzy c-mean and its variants [5–7], deep convolutional neural networks [8], and graph cut [9]. Among the existing image segmentation methods, image thresholding is one of the most popular techniques due to its superiority in simplicity, robustness, and efficiency [10].

Image thresholding works based on the existing information in the image histogram. The histogram shows the distribution of pixel values. In this method, images could be segmented into different regions using one or more threshold values. Image thresholding is widely used in many applications such as food quality [11], satellite image processing [12], character recognition [13], and medical imaging [14].

Image thresholding techniques are divided into two types of parametric and non-parametric approaches. Parametric approaches assume that each region has a statistical distribution, which endeavors to find an approximation of the parameters of this distribution. Kittler and Illingworth [15] argued that a histogram is a mixture of normal distribution. Then, they tried to find the minimum probability of classification error. In another project, Wang et al. [16] integrated histogram information with the spatial information by using Parzen window technique that estimates the unknown probability density function. Dirami et al. [17] estimated the histogram by a weighted sum of Heaviside functions. They improved their method using an enhanced version of the multilevel set method. Parametric approaches have some disadvantages such as computational complexity, time-consumingness, and changing performance based on the image quality [18].

Non-parametric approaches optimize an objective function. Among different objective functions, entropy criterion has attracted the attentions of many researchers [19]. Pun [20] introduced a new image thresholding method based on entropy. Afterward, the method is corrected by Kapur et al. [21] due to some errors in Pun's method. In addition, Kapur et al. [21] introduced a new approach based on the entropy of histogram. The remarkable performance of Kapur's method was studied by the researchers [19, 22, 23] in comparison to other techniques. In [24], researchers reported that the performance of Kapur's method is superior to other entropy-based criteria on the non-destructive images.

From another perspective, image thresholding techniques can be divided into two groups of bi-level thresholding and multilevel thresholding. Bi-level thresholding segments an image into two classes by selecting one threshold value. The first class shows the object and the second one represents the background. Multilevel thresholding divides an image into several regions. The simplest type of thresholding is bi-level thresholding because it selects only one threshold value. However, this problem gets more complicated when the number of thresholds is increased. In fact, the computational complexity is exponential, which leads to a long computation time by the increase of threshold values.

To overcome this drawback, population-based metaheuristic (P-metaheuristic) algorithms can be used as a powerful approach for solving the optimization problems. The P-metaheuristic algorithms are problem-independent algorithms with stochastic operators. These algorithms improve the population of solutions using an iterative process. First, a population of solutions is generated randomly. Then, a new population is generated using some other operators. Finally, the replacement operators integrate the new population with the old population using some selection operators. This process is stopped when some conditions are satisfied.

Two important common questions are posed concerning all P-metaheuristic algorithms: The representation of solutions and the definition of the objective function. The answer to these questions is related to a particular problem. The representation encodes a solution. It has a fundamental role in problem performance. The objective function assigns each solution to a real value number, which describes the quality of the solution. The objective function has a significant effect on the efficiency of P-metaheuristic algorithms since it directs the search process toward the promising solutions of the search space.

Two main criteria in P-metaheuristic algorithms are diversification and intensification. Diversification refers to finding different promising regions of search space, whereas intensification is the process of searching around the best solutions. These two criteria are in conflict with each other and balancing these two criteria is so important in a P-metaheuristic algorithm.

According to the "No Free Lunch" theorem, there is no best algorithm to solve all optimization problems. As a result, various P-metaheuristic algorithms have been studied for image segmentation. Genetic algorithm (GA) is the most well-known P-metaheuristic algorithm. GA has been successfully applied in image thresholding. Yin [25] proposed a new thresholding algorithm based on GA and a learning strategy. Each chromosome coded as a binary string. In addition, learning strategy could accelerate the movement of chromosomes toward the optimal solution. In another work [26], a three level thresholding algorithm was presented based on probability partition, fuzzy partition, entropy theory, and genetic algorithm. They introduced a new fuzzy entropy using probability analysis. They used GA for finding the optimal combination of fuzzy parameters.

Particle swarm optimization (PSO) is another P-metaheuristic algorithm that is used extensively for image segmentation. Maitra and Chatterjee [27] proposed PSO combined with cooperative learning and comprehensive learning for image thresholding. They applied an entropy-based criterion as the cost function. In another research [28], another variant of PSO, which uses social and momentum components of the velocity equation of PSO, was applied for image thresholding. Between-class variance was applied as the cost function. This algorithm is compared with GA, Gaussian-smoothing method, and symmetry-duality method on two images. Gao et al. [10] applied quantum-behaved PSO for image thresholding. They evaluated it on seven images and compared it with some popular algorithms namely, OTSU, ACO, GA-L, PSO, QPSO, and HCPSO. Two strategies of adaptive inertia and adaptive population were used in another study for PSO based image thresholding [29]. Adaptive inertia helped PSO enhance the search efficiency and convergence speed. Adaptive population assisted PSO to jump out of local optima. They compared the algorithm with GPSO and SGA on 16 images.

In addition to the above-mentioned algorithms, the literature review shows that other P-metaheuristic algorithms have also been applied for image thresholding. Some of the most important ones are differential evolution(DE) [30–32], bacterial foraging algorithm (BFA) [33], bat algorithm (BA) [34], artificial bee colony (ABC) [35–37], bee mating optimization (BMO) [38], electromagnetism optimization (EO) [39], firefly algorithm (FF) [40], and kinetic theory optimization (KCO) algorithm [41]. Moreover, comparative studies using more than one P-metaheuristic algorithm have been studied by some researchers in the literature. Akay [22] compared the performance of ABC and PSO and showed that ABC algorithm outperforms PSO algorithm when the number of threshold values is greater than two. Hammouche et al. [42] compared six metaheuristic algorithms: GA, PSO, DE, ACO, simulated annealing (SA), and tabu search (TS). The experimental results indicated that GA, PSO, and DE outperform ACO, SA, and TS. In another study, Osuna-Enciso et al. [43] made a comparison among PSO, DE, and ABC. The used objective function was based on the mixture of Gaussian functions.

In recent years, several P-metaheuristic algorithms have been developed to optimize a problem. Some of the most popular of them are whale optimization algorithm (WOA) [44] inspired by the social behavior of the whales, grey wolf optimizer (GWO) [45] inspired by the hunting mechanism of grey wolves, cuckoo optimization algorithm (COA) [46] that mimics unusual egg-laying of cuckoos, biogeography-based optimization (BBO) [47] that simulates biogeography in nature, teaching–learning-based optimization (TLBO) [48] that mimics the impact of a teacher on the students, gravitational search algorithm (GSO) inspired by the law of gravity [49], imperialist competitive algorithm (ICA) [50] inspired by imperialist competition among countries, and cuckoo search (CS) [51] that gets the idea from the breeding behavior of some cuckoo species.

This paper focuses on the performance analysis of the above-mentioned recently developed P-metaheuristic algorithms for image thresholding. For this purpose, image thresholding should be converted into an optimization problem. Entropy is used as the cost function of the optimization problem. In addition, to make a more comprehensive comparison, these algorithms were compared with GA, DE, PSO, BA, FF algorithms.

The rest of this paper is organized as follows: Sect. 2 demonstrates image thresholding using WOA, GWO, COA, BBO, TLBO, GSA, ICA, and CS algorithms. The experimental results will be evaluated and compared in the next section. Finally, this work is concluded in Sect. 4.

2 Multilevel thresholding using WOA, GWO, COA, BBO, TLBO, GSA, ICA, and CS algorithms

In this section, multilevel image thresholding is explained using eight recently developed P-metaheuristic algorithm. In the proposed methods, Image thresholding is considered

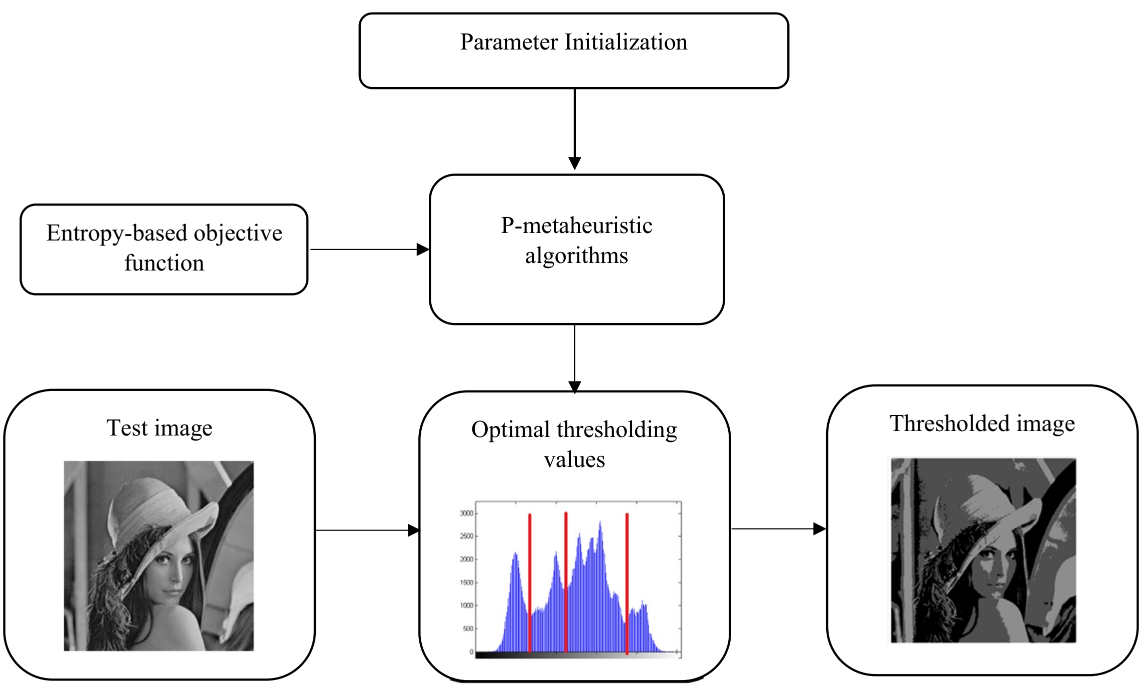

Fig. 1 Block diagram of the proposed methods

as an optimization problem. For this purpose, an entropy-based objective function is applied. The block diagram of the proposed methods is shown in Fig. 1. As illustrated in Fig. 1, first, the parameters of the P-metaheuristic algorithms were initialized. These parameters along with the objective function were fed to the P-metaheuristic algorithms. The output of P-metaheuristic algorithms is the optimal thresholding values. Finally, an image can be segmented by using the thresholding values obtained by P-metaheuristic algorithms. The components of the proposed algorithms were explained with more details in the following subsections.

2.1 The used P-metaheuristic algorithms

This section explains the recently developed P-metaheuristic algorithms. P-metaheuristic algorithms are iterative algorithms to solve optimization problems. These algorithms share some common concepts. All of them use a population of solutions which is initialized with random values. Then, they generate a new population using some operators. Replacement operator carries out a selection using the new and the old populations. This process is stopped when some conditions are satisfied. The general template of P-metaheuristic algorithms is given in Fig. 2. Different P-metaheuristic algorithms have different ways of generation and replacement which leads to different performances.

Each P-metaheuristic algorithm has three elements namely representation, objective function, and operators which are explained as follows:

- Representation: one of the common elements of each P-metaheuristic algorithm is representation that encodes a solution. It is one of the most important characteristics which plays a major role in the efficiency of any P-metaheuristic algorithms.
- Objective function: it assigns a real value to each solution which shows the quality of a solution. The objective function directs the search process toward the good solutions calculated using the objective function. Hence, a proper objective function can have a significant impact on the performance of P-metaheuristic algorithms.
- Operators: each P-metaheuristic algorithm has a number of operators to generate new solutions and select solutions for next iteration. Each P-metaheuristic algorithm uses a different strategy to generate new solutions which leads to different performances.

In this paper, eight recently developed P-metaheuristic algorithms were evaluated for image thresholding. The same representation and objective function were applied for all used P-metaheuristic algorithms. The used P-metaheuristic algorithms are whale optimization algorithm (WOA), grey wolf optimizer (GWO), cuckoo optimization algorithm (COA), biogeography-based optimization (BBO), teaching–learning-based optimization (TLBO), gravitational search algorithm (GSA), imperialist competitive algorithm (ICA), and cuckoo search (CS). In addition, these algorithms were compared with five other well-known P-metaheuristic algorithms. In the following, the recently developed P-metaheuristic algorithms will be explained.

2.1.1 *Whale optimization algorithm*

Whale optimization algorithm (WOA) [44] is one of the latest P-metaheuristic algorithms inspired by the social behavior of the whales. The WOA algorithm consists of three basic operators, which are encircling prey, bubble-net attacking, and searching for prey. These operators are briefly explained below.

2.1.1.1 Encircling prey This operator updates the position of each agent in the neighborhood of the current best solution. In this operator, the best solution is assumed as the target prey. Then, other search agents (the whales) update their position toward the target prey by the following equations:

$$\overline{D} = \left|\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X^*} - \vec{A}.\vec{D} \tag{2}$$

where t is the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X^*}$ is the best solution, and $\vec{X}$ is the position vector

Fig. 2 The general template of P-metaheuristic algorithms

X_0=generate an initial population of solutions
t=0
repeat
  $X_t^{'}$ = generate a new population
  X_{t+1}=replacement-operator($X_t \cup X_t^{'}$) // select the new population
  t=t+1
Until some conditions satisfied
Output: Best solution obtained.

of the search agent, | | is the absolute value, and (.) is the element-by-element multiplication.

The vectors $\vec{A}$ and $\vec{C}$ are calculated using the following equations:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

where $\vec{a}$ is reduced from 2 to 0 and $\vec{r}$ is a random vector between 0 and 1.

2.1.1.2 Bubble-net attacking Bubble-net attacking operator enhances the exploitation of the algorithm. This operator is composed of two parts, and each part is chosen based on a probability of 0.5. These parts are shrinking encircling and spiral updating position, which are briefly explained as follows.

- Shrinking encircling: it is obtained by decreasing the value of $\vec{a}$ in Eq. 3, which leads to exploration in the early iterations of the algorithm and exploitation in the later iterations.
- Spiral updating position: this part calculates the distance between the whale and prey (the best position). Then, a spiral equation is achieved to update the current position as follows:

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) \tag{5}$$

where $\overrightarrow{D'}$ is the distance from the search agents to the prey, b is a constant, l is a random number between −1 and 1, and (.) is the element-by-element multiplication.

One of these two operators, shrinking encircling and spiral updating position, will be selected with a probability of 0.50 as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*} - \vec{A}.\vec{D} & if p < 0.50 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & if p \geq 0.50 \end{cases} \tag{6}$$

2.1.1.3 Search for prey This operator emphasizes on exploration. A mechanism, based on the variation of the $\vec{A}$ vector, is applied to search the prey. To this purpose, $\vec{A}$ with the values greater than 1 or less than −1 is used to get away from a whale. It is modeled as follows:

$$\vec{D} = |\vec{C}.\overrightarrow{X_{rand}} - \vec{X}| \tag{7}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{8}$$

where $\overrightarrow{X_{rand}}$ is a random position vector selected from the current population, $\vec{A}$ and $\vec{C}$ are coefficient vectors, | | is the absolute value, and (.) is the element-by-element multiplication.

The pseudo code of the WOA algorithm is presented in Fig. 3.

Fig. 3 Pseudo code of the WOA algorithm

```
Input: maxI=maximum iteration, N: the number of agents, X: a random population of
solutions, f: the matrix of objective function values
Output: X*

X* =the best search agent
While (t<maxI)
for each search agent
      Update a, A, C, l, and p
      if (p<0.5)
             if(|A|<1)
                  Update the position of current search agent by the Eq.2
            else if (|A|>1)
                  Select a random search agent (x_rand)
                  Update the position of current research agent by the Eq.8
            end-if
      else if (p≥0.5)
            Update the position of the current search by the Eq.5
     end-if
end-for
Calculate the fitness of each search agent and update matrix f
Update X* if there is a better solution
t=t+1
end-while
```

2.1.2 Grey wolf optimizer

Grey wolf optimizer (GWO) [45] is a new P-metaheuristic algorithm which has attracted a great attention in solving optimization problems [52–54]. The GWO algorithm is inspired by the hunting mechanism of grey wolves. Grey wolves live in a hierarchy, which are defined as follows:

- Alpha: it is the leader who is responsible for decision-making about hunting, sleeping, time to wake, and so on.
- Beta: they help the alpha in decision-making. Beta is the best alternative to the alpha.
- Omega: this group plays the role of scapegoat. They are the last group that can eat.
- Delta: if a wolf is not alpha, beta, and omega, it is called delta.

In the GWO algorithm, the best solution is alpha. Subsequently, the second and the third best solutions are beta and omega. The rest of solutions are assumed delta. The operators of the GWO algorithm are briefly demonstrated below.

2.1.2.1 Encircling prey Grey wolves encompass the prey during the hunt. The following equations are used for modeling of encircling behavior:

$$\vec{D} = \left|\vec{C}.\vec{X}_p(t) - \vec{X}(t)\right| \tag{9}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{10}$$

where $\vec{A}$ and $\vec{C}$ are two coefficient vectors, $\vec{X}(t)$ is the grey wolf (best solution), $\vec{X}_p$ is the position vector of the prey, and t is the current iteration. The vectors $\vec{A}$ and $\vec{C}$ can be calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{11}$$

$$\vec{C} = 2.\vec{r_2} \tag{12}$$

where $\vec{a}$ is decreased from 2 to 0 and, $\vec{r_1}$ and $\vec{r_2}$ are random vectors between 0 and 1.

2.1.2.2 Hunting The hunt is guided by the alpha and sometimes the beta and the alpha. There is no information about the location of the prey. In the GWO algorithm, it is assumed that the alpha, beta, and delta have more experience about the location of prey and the delta wolves try to update their positions according to alpha, beta, and delta. The following formulas are used for this purpose:

$$\vec{D}_\alpha = |\vec{C_1}.\vec{X_\alpha} - \vec{X}|,\ \vec{D}_\beta = |\vec{C_2}.\vec{X_\beta} - \vec{X}|,\ \vec{D}_\delta = |\vec{C_3}.\vec{X_\delta} - \vec{X}| \tag{13}$$

$$\vec{X}_1 = \vec{X_\alpha} - \vec{A}_1.(\vec{D_\alpha}),\ \vec{X}_2 = \vec{X_\beta} - \vec{A}_2.(\vec{D_\beta}),\ \vec{X}_3 = \vec{X_\delta} - \vec{A}_3.(\vec{D_\delta}) \tag{14}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{15}$$

2.1.2.3 Attacking prey In order to model the attacking plan to the prey, the value of $\vec{a}$ in Eq. 11 is gradually decreased. It enhances the exploitation of the algorithm.

Fig. 4 Pseudo code of the GWO algorithm

```
Input: maxI=maximum iteration, N: the number of agents, X= a random population of solutions, f: the matrix of objective function values

Output: X*
X_a =the best search agent
X_b =the second best agent
X_c =the third best agent
While(t<maxI)
    for each search agent
        update the position of the current search agent by the Eq.15
    end-for
  Update a, A, and C
  Calculate the fitness of each search agent
  Update X_a , X_b , and X_c
  t=t+1
end-while
X*=the best search agent
```

2.1.2.4 Search for prey Grey wolves get away from each other to search for the prey. To this end, $\vec{A}$ with random values greater than 1 or less than −1 is used. It leads to deviation of search agents from the prey which increases the exploration of the algorithm.

The pseudo code of the GWO algorithm is presented in Fig. 4.

2.1.3 Cuckoo optimization algorithm

Cuckoo optimization algorithm (COA) [46] is another P-metaheuristic algorithm inspired by unusual egg-laying of cuckoos. Recently, the COA algorithm has presented a satisfactory performance in many applications [55–57]. It is important to note that the COA algorithm is different from cuckoo search (CS) [51]. The components of the COA algorithm are briefly described as follows.

2.1.3.1 Definition of egg laying radius Cuckoos lay eggs within a maximum distance from their habitat. In the COA algorithm, each cuckoo has a parameter, which is called "Egg laying radius (ELR)", and is calculated as follows:

$$ELR = \alpha \times \frac{C}{T} \times (h - l) \tag{16}$$

where α is an integer constant, C is the number of current cuckoo's eggs, T is the total number of eggs, h is the upper limit, and l is the lower limit of variables.

2.1.3.2 Cuckoos' style for egg laying In this step, each cuckoo lays eggs randomly in some other host birds' nests within its ELR. After egg laying, eggs with fewer profits will be destroyed.

2.1.3.3 Immigration of cuckoos Communities are constructed when young cuckoos grow up. K-means algorithm is used to construct the communities. Then, mean profit value of each community is calculated. The best mean profit value determines the goal point. Other cuckoos move toward the goal point. Each cuckoo moves λ% of distance between its position and the goal point and has a deviation of ϕ radians.

2.1.3.4 Removing cuckoos in the worst habitats Because of the population balance of cuckoos, some cuckoos with the best profit values survive while others die.

The pseudo code of the COA is shown in Fig. 5.

2.1.4 Biogeography-based optimization

Biogeography-based optimization (BBO) is a well-known P-metaheuristic algorithm, which simulates biogeography in nature. Like other P-metaheuristic algorithms, each solution (habitat) consists of several components. Each component of the solution vector is called SIV. The quality of each solution is measured with HIS. HIS is analogous to "fitness" in the genetic algorithm. High HIS shows habitats with many species and low HIS refers to habitats with few species. There are two significant operators in this algorithm:

2.1.4.1 Migration There are emigration and immigration rates for each solution that represent the possibility of information sharing among habitats. Emigration and immigration rates are dependent on the number of species in the habitat such as the linear curves in Fig. 6.

The emigration rate of each solution is used to modify each SIV by sharing information within the population. If a given SIV in a given solution S is selected for immigration, then the emigration solution k is chosen based on the emigration rate. Migration can be shown as follows:

$$S_i \leftarrow K_j \tag{17}$$

Fig. 5 Pseudo code of the COA algorithm

Input: maxI: maximum iteration, N: the number of agents, X: a random population of cuckoo habitats, f: the matrix of profit values, Nc: the number of clusters
Output: X*
while(t<maxI)
    Dedicate some eggs to each cuckoo
    Let cuckoos to lay eggs inside their corresponding ELR
    Kill those eggs that are recognized by host birds
    Evaluate the habitat if each newly grown cuckoo
    Limit cuckoo's maximum number in environment and kills those who live in worst habitats
    Cluster the population of cuckoo habitats
    Find the best community
    Select goal habitat
    Let new cuckoo population move toward goal habitat
end-while
X*=the best habitat

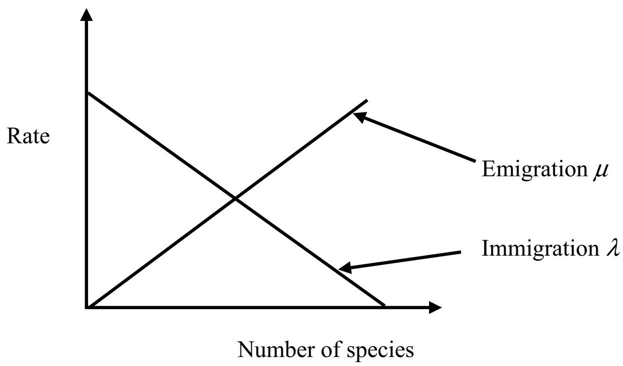

Fig. 6 The relationship between emigration and immigration rates and the number of species in the BBO algorithm

where S_i is the ith SIV in the solution S and K_j is the jth SIV in the solution K.

2.1.4.2 Mutation Mutation is another operator, which randomly changes SIV. It increases diversification of the algorithm.

The pseudo code of the BBO algorithm is presented in Fig. 7.

2.1.5 Teaching–learning-based optimization

Teaching–learning-based optimization (TLBO) is a new P-metaheuristic algorithm which is presented competitive results in many applications [58, 59]. The TLBO algorithm mimics the impact of a teacher on students. In this algorithm, each candidate solution is called a student. In addition, the best solution is intended as the teacher. The score is analogous to "fitness" in the genetic algorithm. A good teacher can increase the mean scores in one class. The TLBO algorithm has two phases: teacher phase and student phase. These two phases are clarified below.

2.1.5.1 Teacher phase Let M_i be the mean value of the scores and T_i be the teacher at iteration i. T_i will try to transfer the mean value of the scores to the new mean, and the new mean is better than the old one. For this purpose, the TLBO algorithm proposes the following formula:

$$Difference\_Mean_i = r_i(M^* - T_F M_i) \tag{18}$$

Fig. 7 Pseudo code of the BBO algorithm

```
Input: maxI:maximum iteration, N: the number of agents, X: a random population of
habitats, f: the matrix of HSI values
Output: X*

while(t<maxI)
  Calculate the immigration rate λ_i and the emigration rate μ_i for each habitat based on
HSI.
  for i from 1 to N
      Select X_i with probability proportional to λ_i
      if rand (0,1)< λ_i
          for j from 1 to N
              Select X_j with probability proportional to μ_i
              if rand(0,1)< μ_i
                  Randomly select a variable x from X_j
                  Replace the corresponding variable in X_i with x
                  break for j
              end-if
          end-for
      end-if
  end-for
  for i from 1 to N
      Calculate the probability of each habitat based on λ_i and μ_i
  end-for
  Select X_i with respect to p_i
  if rand(0,1)<mutation rate
      Replace a random variable x from X_i with a randomly generated number
  end-if
  Recompute HIS values
end-while
X*=the best habitat
```

$$X_{new,i} = X_{old,i} + Difference\_Mean_i \quad (19)$$

where r_i is a random number between 0 and 1, T_F is a constant, M^* is the score of the teacher, M_i is the mean scores, $X_{old,i}$ is the ith dimension of current solution, and $X_{new,i}$ is the ith dimension of the new solution.

2.1.5.2 Student phase In this step, students communicate with each other to increase their knowledge. A student can learn new contents from other more knowledgeable students. The following formula is used for this purpose:

$$X_{new} = X_i + r(X_i - X_j) \quad (20)$$

where r is a random number between 0 and 1, and, X_i and X_j are two random students, where X_j has a higher score than X_i.

The pseudo code of the TLBO algorithm is presented in Fig. 8.

2.1.6 Gravitational search algorithm

Gravitational search algorithm (GSA) is another well-known recently developed P-metaheuristic algorithm. The GSA algorithm has attracted much attentions in different applications [60–62]. It is inspired by the law of gravity. In the real world, all objects attract each other by the gravity force. Each candidate solution is called a mass. Each mass has four characteristics: position, inertia mass, active gravitational mass, and passive gravitational mass. The position of a mass corresponds to a solution, and its gravitational and inertial masses are determined using the fitness function. The GSA algorithm has used two laws for optimization: gravity, and motion. These two laws are explained below.

2.1.6.1 Law of gravity Each mass attracts other masses. The force acting on mass i from mass j is defined as follows:

$$F_{ij}^d(t) = G(t).\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij} + \varepsilon}(x_j^d(t) - x_i^d(t)), \quad (21)$$

where F_{ij}^d is the force acting on mass i from mass j in the dth dimension, G is the gravitational constant, M_{pi} is the passive gravitational mass related to mass i, M_{aj} is the active gravitational mass related to mass j, R_{ij} is the

Fig. 8 Pseudo code of the TLBO algorithm

Input: maxI=maximum iteration, N: the number of agents, X= a random population of solutions, f: the matrix of objective function values
Output: $X_{teacher}$
$X_{teacher}$=the best solution
while(t<maxI)
    for i from 1 to N
        M_D =calculate the mean of variables in the population
        M^* =score of $X_{teacher}$
        X_{new} =Update the position of ith solution using Eqs. 18 and 19
        if score of X_{new} is better than score of X_i
            $X_i = X_{new}$
        end-if
        Select two different students X_i and X_j randomly
        if score of X_i is better than X_j
            $X_{new} = X_i + r(X_i - X_j)$
            if the score of X_{new} is better than X_i
                $X_i = X_{new}$
            end-if
        else-if
            $X_{new} = X_j + r(X_j - X_i)$
            if the score of X_{new} is better than X_j
                $X_j = X_{new}$
            end-if
        end-if
    end-for
    $X_{teacher}$=the best solution
end-while

Euclidian distance between two masses i and j, x_i^d is the position of ith mass in the dth dimension, x_j^d is the jth mass in the dth dimension, and t is the current iteration.

The total force on mass i in the dimension d is calculated according to the following equation:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t), \tag{22}$$

where F_{ij}^d is the force acting on mass i from mass j, N is the number of masses, and $rand_j$ is a random number between 0 and 1.

The gravitational constant is a reduction constant. It is reduced along with the time to control the search accuracy. Moreover, the inertial mass is calculated by the following equations:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{23}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{24}$$

where M_i is the inertial mass, fit_i is the fitness of mass i, *worst* is the worst fitness among masses, and *best* is the best fitness among masses.

2.1.6.2 Law of motion In this step, the new position of a mass is calculated as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{25}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{26}$$

where $rand_i$ is a random number between 0 and 1, v_i^d is the velocity of mass i in the dth dimension, and a_i^d is the acceleration of the mass i in the dth dimension. The acceleration is calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{27}$$

The pseudo code of the GSA algorithm is presented in Fig. 9.

2.1.7 Imperialist competitive algorithm

Another P-metaheuristic algorithm studied in this paper is imperialist competitive algorithm (ICA) [50], which mimics imperialist competition among countries. Each solution in this algorithm is called country. The power of a country shows the quality of a solution. The components of the ICA algorithm are presented in the following.

2.1.7.1 Empire establishment In this step, countries are divided into two categories: imperialists and colonies. The most powerful countries establish imperialists, while the remaining countries are colonies. Each empire has an imperialist and some colonies. Colonies are divided among imperialists based on their power. The number of colonies of an empire is calculated as follows:

$$C_n = c_n - \max_i \{c_i\} \tag{28}$$

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \tag{29}$$

$$N.C_n = round\{p_n.N_{col}\} \tag{30}$$

where c_n is the cost of nth imperialist, C_n is the normalized cost, P_n is the normalized power of each imperialist, N_{col} is the number of colonies, and $N.C_n$ is the initial number of colonies of nth empire.

2.1.7.2 Assimilation Assimilation operator transfers colonies toward the imperialist by x units. The variable of x is a random variable with uniform distribution described by the following equation:

$$x \sim U(0, \beta \times d) \tag{31}$$

where β is a number greater than 1, and d is the distance between the colony and the imperialist.

Fig. 9 Pseudo code of the GSA algorithm

```
Input: maxI=maximum iteration, N: the number of agents, X= a random population of
solutions, f: the matrix of objective function values
Output: X*
X*=the best solution
while(t<maxI)
  Update G, best, and worst of the population
   Calculate M and a for each agent using Eqs. 24 and 27
   Update velocity and position using Eqs. 25 and 26
end-while
```

2.1.7.3 Exchanging the position of the imperialist and the colony If a colony reaches a better position than that of the imperialist, the position of colony and imperialist will be exchanged.

2.1.7.4 Total power of an empire The total power of an empire is calculated as follows:

$$T.C_n = Cost(imperialist_n) + \xi mean\{Cost(coloniesofempire_n)\} \tag{32}$$

where $T.C_n$ is the total cost of nth empire, and ξ is a positive number less than 1.

2.1.7.5 Imperialist competition All empires try to occupy colonies of other empires to increase their power. For this purpose, all empires start to compete to possess the weakest colony of the weakest empire. Empires that are more powerful have more chance to possess this colony.

The pseudo code of the ICA algorithm is shown in Fig. 10.

2.1.8 Cuckoo search

Cuckoo search (CS) [51] is a P-metaheuristic algorithm inspired by the breeding behavior of some cuckoo species. As mentioned previously, the CS algorithm uses a different strategy compared with the COA algorithm. However, both of them mimic the unusual behavior of cuckoos. The CS algorithm uses three rules to optimize a problem: (1) each cuckoo lays an egg which will be dumped in a randomly selected nest, (2) the best nests with the best eggs will be kept to the next iteration, and (3) the number of nest hosts is a constant number in the whole process of search.

In this algorithm, each egg is equivalent to a solution. Each new solution is generated using Eq. 33.

$$x_{new} = x_{old} + \alpha \oplus Levy \tag{33}$$

where α is the step size, $Levy$ shows Levy flight distribution, and $\oplus$ means entry-wise multiplications. A Levy flight is a type of random walk that step length is drawn from a Levy distribution.

After generating a new solution, it will be compared with the old solution. In case the objective value of the new

Fig. 10 Pseudo code of the ICA algorithm

```
Input: maxI=maximum iteration, N: the number of agents, X= a random population of
countries, f: the matrix of objective function values
Output: X*
X*=the best solution
Establish empires
while(t<maxI)
        Move the colonies toward the corresponding imperialist
        if there is a colony with the better cost of its imperialist
                Exchange the position of the imperialist and the colony
        end-if
        Compute the total cost of each empire using Eq.32
        Do imperialist competition
        Remove the weak empires
end-while
```

Fig. 11 Pseudo code of the CS algorithm

```
Input: maxI=maximum iteration, N= the number of agents, X= a random population of
solutions, f: the matrix of objective function values
Output: X*
X*=the best solution
while(t<maxI)
  Get a cuckoo randomly by Levy flight
  Evaluate the objective function of each cuckoo
  Choose a nest among n(say,j) randomly
  if(Fi>Fj)
        Replace J by the new solution
  end
  Remove a fraction of worse nests and build new ones
  Keep the best solution
  X*=Find the best solution
end
```

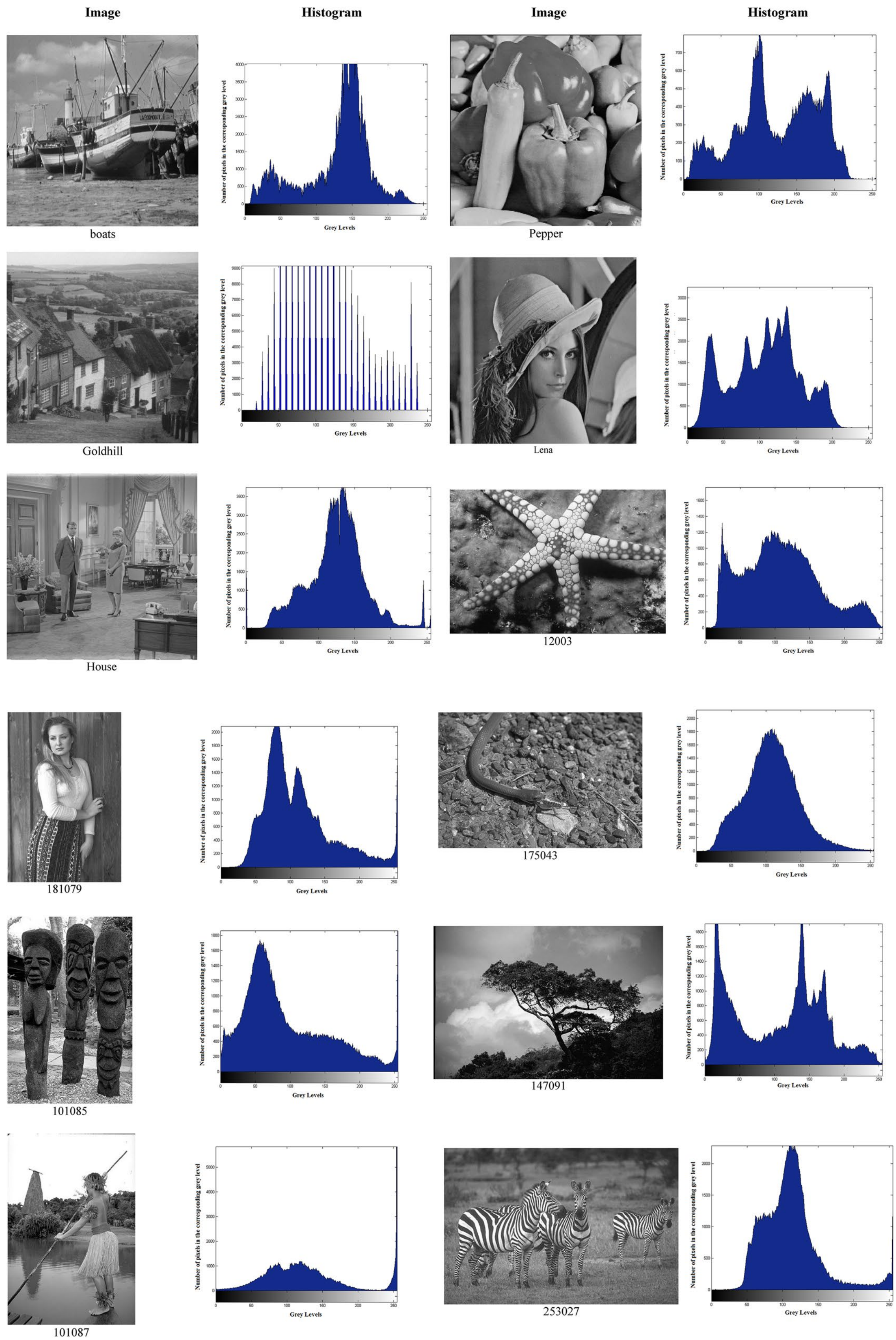

Fig. 12 Benchmark images and the corresponding histograms

Table 1 Default parameter settings

Algorithms	Parameters	Value
WOA	A constant for defining the shape of the logarithmic spiral (b) [44]	1
GWO	No parameter	–
COA	Number of groups [46]	3
	Minimum number of eggs [46]	5
	Maximum number of eggs [46]	20
BBO	Habitat modification probability [64]	1
	Maximum immigration rate [64]	1
	Maximum emigration rate [64]	1
TLBO	No parameter	–
GSA	No parameter	–
ICA	Number of empires [65]	5
	Coefficient associated with average power [65]	0.1
	Revolution rate [65]	0.2
	Deviation assimilation parameter [65]	$\pi/4$
	Direction assimilation parameter [65]	0.5
CS	The probability of finding alien eggs (P_a) [51]	0.25
	Step size taken by cuckoos [51]	1
BA	Loudness [66]	0.7
	Pulse rate [66]	0.7
FF	Light absorption coefficient (γ) [67]	1
	Attractiveness at r = 0 (β_0) [67]	1
	Scaling factor (α) [67]	0.2
PSO	Cognitive constant (C1) [68]	2
	Social constant (C2) [68]	2
	Inertia constant (w) [68]	1 to 0
DE	Scaling factor [69]	0.5
	Crossover probability [69]	0.1
	Selection method [69]	DE/rand/1/bin
GA	Crossover probability	0.9
	Mutation probability	1/(the number of thresholds)
	Selection method	Roulette wheel

solution is better than the old solution, the new solution will be replaced by the old solution. The pseudo code of the CS algorithm is demonstrated in Fig. 11.

2.2 Representation strategy

The goal of multilevel image thresholding is to find $m(m \geq 2)$ proper threshold values to segment an image. The output image can be obtained using Eq. 34.

$$\begin{aligned} M_0 &= \{g(x,y) \in I \mid 0 \leq g(x,y) \leq t_1 - 1\} \\ M_1 &= \{g(x,y) \in I \mid t_1 \leq g(x,y) \leq t_2 - 1\} \\ M_i &= \{g(x,y) \in I \mid t_i \leq g(x,y) \leq t_{i+1} - 1\} \\ M_m &= \{g(x,y) \in I \mid t_m \leq g(x,y) \leq L - 1\} \end{aligned} \tag{34}$$

where $t_i (i = 1, \ldots, m)$ is the ith threshold value, and m is the number of threshold values.

Representation is one of the main components of each P-metaheuristic algorithm, which encodes a candidate solution. It has a great impact on the performance of the algorithm. Candidate solution has different names in different algorithms such as chromosome in GA, wolf in GWO, habitat in BBO, and country in ICA. In this paper, a real-value coding is used for image thresholding. The length of each array is m, which is the number of threshold values. This array is defined as follows:

$$A\ candidate\ solution = \left[t_1, t_2, \ldots, t_{i+1}, \ldots, t_m\right] \tag{35}$$

Table 2 The obtained objective function values for benchmark images

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	12.5733	12.5747	12.5747	**12.5748**	**12.5748**	**12.5748**	**12.5748**	**12.5748**	12.5718	**12.5748**	12.574	12.5746	12.557
	3	15.8187	15.8206	15.8194	**15.8209**	**15.8209**	15.8207	**15.8209**	**15.8209**	15.8156	**15.8209**	15.8151	15.8176	15.7754
	4	18.6485	18.6548	18.6533	18.6532	18.6552	18.6281	18.6486	**18.6557**	18.6055	18.6549	18.6321	18.628	18.5152
	5	21.3831	21.3771	21.3825	20.7412	21.395	21.4003	**21.4015**	21.4014	21.3462	21.3995	21.3273	21.344	21.2258
Pepper	2	**12.5198**	**12.5198**	**12.5198**	**12.5198**	**12.5198**	**12.5198**	**12.5198**	12.5198	12.5193	**12.5198**	12.5198	12.5196	12.5134
	3	15.5872	**15.5874**	15.5871	15.5872	**15.5874**	**15.5874**	**15.5874**	**15.5874**	**15.5874**	15.5874	15.5854	15.5833	15.5465
	4	18.3527	18.3501	18.3515	18.3497	18.3529	18.3517	18.3549	**18.3551**	18.3327	18.3521	18.3396	18.3344	18.287
	5	21.0419	21.0586	21.041	20.7412	21.0546	21.0432	**21.0592**	21.0589	20.9935	21.0368	21.0012	21.0038	20.9338
Goldhill	2	8.5378	**9.0613**	**9.0613**	**9.0613**	**9.0613**	8.8373	8.8962	**9.0613**	8.44	8.3962	9.0613	9.0077	8.5966
	3	10.7403	**10.9874**	10.9691	**10.9874**	**10.9874**	10.7757	10.7102	**10.9874**	10.6246	10.4105	10.976	10.9529	10.5283
	4	12.3596	12.483	12.4366	12.8468	12.6251	11.9452	12.1693	12.7512	12.2317	11.1793	12.4589	12.4425	12.1919
	5	13.5217	13.7996	13.7881	13.7412	14.0917	13.3004	13.8013	**14.2591**	13.3699	13.798	13.7985	14.0274	13.7183
Lena	2	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	**12.3545**	12.3543	12.354	12.3265
	3	15.6653	15.6862	15.6856	15.6865	**15.6867**	15.5987	15.3257	15.6867	15.6379	15.3257	15.6672	15.6604	15.4634
	4	18.6261	18.6577	18.6485	18.6539	**18.6579**	18.5232	18.0251	18.6578	18.5295	18.0214	18.6269	18.6235	18.4221
	5	21.1152	21.3354	21.3096	20.7412	21.3524	21.1397	20.6259	**21.3569**	21.2145	20.6255	21.2314	21.2955	21.0109
House	2	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	**12.3703**	12.37	**12.3703**	12.3594
	3	15.3688	15.3684	15.3678	15.3688	**15.3695**	15.3584	**15.3695**	**15.3695**	15.3543	15.3694	15.3644	15.3567	15.3181
	4	18.2712	18.2712	18.262	18.2722	18.2749	18.2718	**18.2765**	**18.2765**	**18.2675**	18.2751	18.2535	18.2552	18.1128
	5	20.9088	20.9067	20.907	20.7412	20.917	**20.9195**	**20.9195**	20.9194	**20.9195**	20.9185	20.8564	20.8659	20.7872
12,003	2	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	**12.999**	12.9989	12.9914
	3	16.1661	16.1661	16.166	16.166	**16.1663**	**16.1663**	**16.1663**	**16.1663**	**16.1663**	**16.1663**	16.164	16.1632	16.1273
	4	19.118	19.1179	19.1174	19.1166	19.1183	**19.1184**	**19.1184**	**19.1184**	19.1181	19.1183	19.1093	19.1048	19.0516
	5	21.8756	21.8728	21.8724	20.7412	21.8761	**21.8768**	21.8767	21.8767	21.8724	21.8765	21.8528	21.8361	21.765
181,079	2	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	**12.4912**	12.4911	12.4907	12.4833
	3	**15.5987**	15.5985	15.5986	15.5986	**15.5987**	**15.5987**	**15.5987**	**15.5987**	15.5915	**15.5987**	15.5949	15.5947	15.5564
	4	18.4726	18.4711	18.4704	18.4704	**18.4731**	18.473	**18.4731**	**18.4731**	18.4372	18.473	18.4571	18.4597	18.3636
	5	21.1421	21.1407	21.1436	20.7412	21.1474	21.1474	21.1479	**21.1481**	21.1364	21.1468	21.0895	21.1121	21.0123
175,043	2	12.434	**12.4341**	**12.4341**	**12.4341**	**12.4341**	**12.4341**	**12.4341**	**12.4341**	12.4329	**12.4341**	**12.4341**	12.434	12.4272
	3	15.6939	15.6941	15.694	15.6939	**15.6946**	**15.6946**	**15.6946**	**15.6946**	15.6847	**15.6946**	15.689	15.6864	15.6698
	4	18.7112	18.7116	18.7108	18.7097	**18.7131**	**18.7131**	**18.7131**	**18.7131**	18.7035	18.713	18.6991	18.6977	18.6294
	5	21.4985	21.5006	21.4981	20.7412	21.5024	**21.5051**	21.5048	21.5046	21.4974	21.5042	21.4542	21.4625	21.371
101,085	2	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	**12.9904**	12.9903	12.9743
	3	16.1522	**16.1524**	16.1522	16.1523	**16.1524**	**16.1524**	**16.1524**	**16.1524**	16.1478	**16.1524**	16.1504	16.1453	16.1182
	4	19.202	19.2016	19.2019	19.2013	**19.2024**	**19.2024**	**19.2024**	**19.2024**	**19.2024**	**19.2024**	19.1921	19.1863	19.1435
	5	21.9768	**21.9784**	21.9749	20.7412	21.9772	**21.9784**	21.9782	21.9782	**21.9784**	21.978	21.941	21.9482	21.8416
147,091	2	12.8899	**12.8902**	**12.8902**	**12.8902**	**12.8902**	**12.8902**	**12.8902**	**12.8902**	**12.8902**	**12.8902**	12.89	12.8901	12.8775
	3	16.2377	**16.2401**	16.2397	**16.2401**	**16.2401**	**16.2401**	**16.2401**	**16.2401**	**16.2401**	**16.2401**	16.2337	16.2322	16.1497
	4	19.1429	19.1447	19.1435	19.1468	19.1478	19.1206	**19.1479**	19.148	19.0889	19.1437	19.1176	19.1228	19.0356
	5	21.9366	21.9307	21.9332	20.7412	21.9384	21.9392	21.94	**21.9417**	21.9052	21.9313	21.882	21.9046	21.7963
101,087	2	**11.3573**	**11.3573**	**11.3573**	11.3571	**11.3573**	**11.3573**	**11.3573**	**11.3573**	11.3565	**11.3573**	11.3568	11.357	11.3391
	3	14.1927	**14.1936**	14.1926	14.1929	**14.1936**	**14.1936**	**14.1936**	**14.1936**	14.1923	**14.1936**	14.1876	14.1879	14.1013
	4	16.7994	**16.801**	16.7917	16.7972	16.8008	16.7673	16.7954	16.8008	16.765	16.7953	16.7618	16.7701	16.6594
	5	19.8427	19.8325	19.8428	20.7412	19.8595	19.8017	**19.8616**	19.8614	19.7975	19.8586	19.8034	19.7825	19.5952
253,027	2	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	**12.2727**	12.2726	12.2511
	3	15.2802	15.2796	15.2799	15.28	**15.2803**	**15.2803**	15.2802	**15.2803**	15.2748	15.2802	15.2781	15.2773	15.2213
	4	18.1498	18.1502	18.1498	18.1499	18.1516	18.1515	18.1515	**18.1517**	18.1277	18.1513	18.1331	18.1369	18.0866
	5	20.7484	20.7426	20.742	20.7412	20.7492	20.7493	**20.7516**	**20.7516**	20.7357	20.7497	20.7131	20.7177	20.6615

Bold values indicate the best results

Table 3 PSNR values of benchmark images for 13 different multilevel thresholding methods

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	14.6361	14.6601	14.6601	**14.6735**	14.6489	14.6699	**14.6735**	**14.6735**	14.6114	**14.6735**	14.5826	14.6417	14.5601
	3	17.6031	17.5688	17.5629	17.5813	17.5401	17.6009	17.5629	**17.6544**	17.6052	17.551	17.5629	17.5646	17.4714
	4	19.4976	19.5093	19.4944	19.4604	19.4838	19.3721	19.4644	**19.5102**	19.0073	19.4331	19.4653	19.2308	18.8769
	5	20.4016	20.4182	20.282	20.1213	20.0312	20.0977	20.4710	**20.4812**	20.0642	20.128	20.4499	20.0799	20.1733
Pepper	2	16.2638	16.2638	16.2652	16.2666	16.2673	16.2638	16.2638	**16.2748**	16.2388	16.2638	16.2666	16.2713	16.1536
	3	18.1833	18.1833	18.1894	18.1883	18.1705	18.1741	18.1833	**18.192**	18.1833	18.1737	18.1833	18.1808	18.1177
	4	20.175	20.2635	20.2571	20.2662	18.8605	20.3297	19.0269	**20.4891**	19.8762	20.0173	20.0479	20.1193	19.8432
	5	20.947	20.9628	21.1109	21.0496	20.8805	20.9636	**21.4598**	20.9894	21.2388	20.9903	21.0227	21.2078	20.8535
Goldhill	2	7.0209	**7.6685**	**7.6685**	**7.6685**	7.6317	7.5094	7.2598	**7.6685**	7.2597	7.3679	**7.6685**	7.6251	7.2987
	3	11.1932	10.8396	10.8851	10.82	10.86	10.8216	10.9306	**10.9434**	10.8118	10.7018	10.2792	10.8551	10.6377
	4	12.3541	12.9794	12.4495	11.0231	12.2938	**12.98**	12.5943	11.0235	11.1294	11.5216	11.7847	10.5814	12.1354
	5	14.6015	**14.8624**	14.3453	10.7065	14.2521	13.8828	14.2521	13.0247	13.0868	14.291	14.0374	12.0721	13.8396
Lena	2	15.7023	**15.7071**	**15.7071**	**15.7071**	**15.7071**	**15.7071**	**15.7071**	**15.7071**	**15.7071**	**15.7071**	15.6949	15.699	15.6572
	3	15.4776	15.6983	15.7045	15.7001	15.7071	15.1958	15.57	**15.7071**	15.0781	15.5801	15.5226	15.493	15.5681
	4	18.3184	18.5604	18.4994	18.5559	18.5802	18.1093	18.57	**18.9027**	18.5666	18.233	18.3083	18.4076	17.9748
	5	18.9209	19.9088	19.8346	19.9864	20.2279	19.7276	**20.3665**	20.2448	19.9439	20.2418	19.7668	20.0699	19.434
House	2	14.48	14.48	14.48	14.48	**14.4958**	14.48	14.48	14.48	14.48	14.4935	14.4722	14.48	14.4117
	3	17.8935	17.9199	17.859	17.9103	17.8959	17.7717	**17.9212**	**17.9212**	17.3252	17.9181	17.8579	17.7061	16.9874
	4	18.7558	18.7607	**19.0005**	18.7846	18.7644	18.8814	18.7772	18.7772	18.7772	18.8398	18.915	18.8104	18.6533
	5	20.7133	20.7804	20.7955	20.7561	20.7597	20.7657	20.7597	**20.8179**	20.7597	20.7958	20.7071	20.6788	20.4138
12,003	2	14.3191	14.3191	14.3191	14.3204	14.1391	**14.3517**	14.3191	14.3191	14.3191	14.3453	14.3217	14.3119	14.121
	3	16.8745	16.8741	16.873	16.8658	16.8639	16.887	16.8639	**16.8956**	16.8639	16.8914	16.8428	16.8639	16.2861
	4	18.1523	18.1606	18.1341	**18.1846**	18.1786	18.1514	18.1514	18.1514	18.133	18.1719	18.1591	18.1248	18.1261
	5	19.9736	20.0011	19.9442	19.9614	**20.0068**	19.987	19.977	19.9535	19.8705	20.0046	19.8852	19.654	19.3878
181,079	2	10.8126	10.8126	10.8126	10.8441	10.8524	**10.8602**	10.8126	10.8126	10.8598	10.86	10.8126	10.8852	10.1228
	3	13.5221	13.5235	13.4838	13.4728	13.5542	13.5229	13.5229	**13.5633**	13.5229	13.5611	13.5255	13.5085	13.066
	4	18.2495	18.2087	18.3383	18.2498	18.2872	18.3393	18.3873	**18.3876**	18.305	18.336	18.2498	18.2498	17.0655
	5	19.4302	**19.5524**	19.4676	19.4413	19.4922	19.5207	19.4906	19.5212	19.4114	19.4785	19.224	19.3716	18.9439
175,043	2	14.6405	14.6461	14.6461	14.6461	**14.6742**	14.6687	14.6461	14.6461	14.5638	14.6461	14.6461	14.6407	14.3615
	3	15.8965	15.8865	15.8876	15.897	**15.9348**	15.9018	15.9018	15.9018	15.7214	15.9281	15.7388	15.7988	15.6453
	4	17.4998	17.4375	17.4511	17.5427	17.4271	17.4627	17.3904	**17.5902**	17.3689	17.5012	17.3689	17.504	17.0745
	5	19.3471	19.3722	19.2668	19.2283	**19.4644**	19.4082	19.3187	19.3682	19.3905	19.4123	19.3261	19.3895	19.3495
101,085	2	13.82	13.8228	13.8228	13.8228	13.8393	**13.8422**	13.8228	13.8228	13.8228	13.8228	13.8241	13.8211	13.8147
	3	17.2622	17.2599	17.26	17.26	17.236	17.2422	**17.2663**	**17.2633**	16.9745	17.2416	17.2618	17.2228	16.5513
	4	18.9083	18.9052	18.8942	18.9065	18.8963	18.8946	**18.9085**	**18.9085**	**18.9085**	18.89	18.8717	18.8619	18.703
	5	**20.1775**	20.1496	20.1443	20.1382	20.1761	20.1318	20.1684	20.1575	20.1443	20.1679	20.1591	20.0806	19.7676
147,091	2	14.5561	**14.5571**	**14.5571**	**14.5571**	**14.5571**	14.5542	**14.5571**	**14.5571**	**14.5571**	14.5554	14.55	14.5546	14.505
	3	17.3426	17.3426	17.3426	17.3435	17.3198	17.3396	17.3426	**17.3656**	17.3426	17.3193	17.3535	17.3382	17.1177
	4	19.1014	19.1082	19.1102	19.1002	19.0909	19.0223	**19.1296**	19.1014	18.5615	19.103	19.0998	19.0681	18.7378
	5	19.6371	19.6247	19.7464	19.5988	19.7835	19.6212	19.7302	**19.8095**	19.613	19.7875	19.7547	19.6644	19.77
101,087	2	**9.2446**	**9.2446**	**9.2446**	**9.2446**	9.2197	9.2142	**9.2446**	**9.2446**	9.2412	9.2181	9.2236	9.2345	9.2092
	3	10.067	10.0754	10.0633	10.0633	10.0427	10.0483	**10.0754**	**10.0754**	**10.0754**	10.05	10.0542	10.0637	10.056
	4	11.4801	11.4601	12.004	11.4864	11.4214	12.6842	12.0228	11.4738	**13.3482**	11.9838	11.4601	11.2227	10.6659
	5	19.6057	19.695	19.7084	19.6755	19.8096	19.5424	**19.8725**	19.8662	19.1539	19.7556	19.6012	19.6676	19.2115
253,027	2	13.747	13.7372	13.7372	13.7372	13.7372	13.7863	13.7372	**13.7962**	13.7372	13.7372	13.7373	13.7567	13.7018
	3	14.9392	14.9301	14.8878	14.8898	**14.972**	14.943	14.9227	14.943	14.8168	14.9362	14.8995	14.8551	14.9099
	4	15.4386	15.5178	15.4731	15.4912	15.5445	15.5315	15.491	**15.5515**	15.5315	15.5233	15.519	15.4093	15.5188
	5	17.241	17.1016	18.2878	17.8633	18.3338	18.4158	18.3085	**18.602**	18.0634	17.9565	17.2003	17.0685	17.6341

Bold values indicate the best results

Table 4 FSIM values of benchmark images for 13 different multilevel thresholding methods

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	0.7536	0.7548	0.7548	0.7549	0.7549	**0.7551**	0.7549	0.7549	0.7537	0.7551	0.7545	0.7547	0.7467
	3	0.8328	0.8327	0.8328	0.833	0.8325	0.8328	0.8328	**0.8332**	0.8297	0.8326	0.833	0.8322	0.8286
	4	0.8669	0.8679	0.8669	0.8664	0.8677	0.8682	**0.8686**	0.8685	0.8651	0.8663	0.8668	0.8625	0.8593
	5	0.8964	0.8978	0.8948	0.8912	0.8892	0.8903	0.8892	**0.8989**	0.8811	0.8909	0.8974	0.8803	0.8864
Pepper	2	0.6902	0.6902	0.6903	0.6905	0.6909	0.6902	0.6902	**0.6911**	0.6902	0.691	0.6905	0.691	0.6901
	3	0.7524	**0.7526**	0.7521	0.7521	0.7519	0.7519	**0.7526**	**0.7526**	**0.7526**	0.752	0.7511	0.7505	0.7464
	4	0.7912	0.7921	0.7925	0.7932	0.7726	**0.7945**	0.7749	0.7727	0.786	0.7895	0.7908	0.7921	0.7872
	5	0.8165	0.8174	0.819	0.8197	0.816	0.8173	0.8174	**0.8197**	0.8197	0.8147	0.8183	**0.811**	0.8109
Goldhill	2	0.5236	0.5621	0.5611	0.5621	0.5621	0.5621	**0.7339**	0.5762	0.5704	0.7336	0.5621	0.5541	0.5321
	3	0.5303	0.5891	0.5986	0.5897	0.5856	0.5897	0.5322	**0.5898**	0.5284	0.5223	0.5984	0.5957	0.5312
	4	0.65	0.6698	0.6561	0.5961	0.6484	0.7315	0.8604	0.6025	0.6361	**0.8604**	0.6283	0.5813	0.6529
	5	0.7241	0.7402	0.7198	0.584	0.7216	0.7006	**0.8941**	0.6611	0.7034	0.8928	0.7128	0.6419	0.699
Lena	2	0.7347	0.7349	0.7349	0.7349	0.7359	0.7349	0.7349	**0.7363**	0.7349	**0.7363**	0.734	0.7346	0.7351
	3	0.7321	0.7346	0.7346	**0.7349**	0.7339	0.7322	0.7322	**0.7349**	0.7343	0.7336	0.7341	0.7305	0.7206
	4	0.8034	0.8016	0.803	0.8018	0.8033	0.7998	**0.8463**	0.802	0.8121	0.8022	0.7996	0.8038	0.8011
	5	0.8256	0.8489	0.8478	0.8482	0.8461	0.8409	**0.8772**	0.8461	0.8449	0.8463	0.8462	0.8473	0.833
House	2	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	**0.7135**	0.7131	0.7135	0.7114
	3	0.8021	0.8032	0.8004	0.8028	0.8026	0.7981	**0.8032**	**0.8032**	0.7951	**0.8032**	0.8015	0.7981	0.7927
	4	0.8364	0.8365	0.8364	0.8369	0.8362	0.8396	**0.8437**	**0.8437**	0.8364	0.8388	0.8411	0.8387	0.8437
	5	0.8852	0.8864	0.8869	0.8865	0.8857	0.8857	**0.8871**	**0.8871**	0.8857	0.8865	0.8858	0.8861	0.8786
12,003	2	0.6047	0.6047	0.6047	0.6049	0.6047	0.6051	0.6047	**0.6052**	0.6047	0.6047	0.605	0.605	0.6025
	3	0.6855	0.6858	0.6855	0.6853	0.6855	0.6861	0.6855	**0.6865**	0.6855	**0.6865**	0.6846	0.6863	0.6702
	4	0.7364	0.7369	0.7354	0.7369	0.7372	0.7365	0.7365	**0.7375**	0.7354	0.7369	0.7362	0.7351	0.7365
	5	0.7897	0.7897	0.7882	0.7901	0.7887	0.7894	0.7894	**0.7902**	0.7868	0.7887	0.7883	0.7826	0.7737
181,079	2	0.5944	0.5944	0.5944	**0.5946**	**0.5946**	0.5945	**0.5946**	**0.5946**	0.5947	0.5945	0.5941	0.5941	0.5801
	3	0.6697	0.6697	**0.6698**	**0.6698**	**0.6698**	**0.6698**	**0.6698**	**0.6698**	0.6697	0.6697	0.6693	0.6691	0.666
	4	0.7204	0.7191	0.7231	0.7249	0.721	0.7222	0.7203	0.7203	0.7213	0.7219	0.7203	0.7239	0.7096
	5	0.762	0.7658	0.7632	0.763	0.7653	0.7654	0.7651	**0.766**	0.76	0.7633	0.7587	0.7613	0.7585
175,043	2	0.6466	0.6474	0.6474	0.6474	**0.6491**	**0.6491**	**0.6491**	**0.6491**	0.6408	0.6474	0.6474	0.6466	0.6399
	3	0.7313	0.7308	0.7308	0.7312	0.7314	0.7314	**0.7331**	**0.7331**	0.7214	0.7328	0.7243	0.7268	0.7183
	4	0.7946	0.7924	0.7926	0.796	0.7928	0.7941	0.7914	0.7909	**0.7971**	0.7951	0.7953	0.7932	0.7777
	5	0.8439	0.8458	0.8441	0.8416	0.8447	**0.8469**	0.8446	0.8455	0.8464	0.8468	0.8462	0.8448	0.838
101,085	2	0.6476	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	**0.6477**	0.6476	0.6412
	3	0.7724	0.7727	0.7727	0.7727	0.7726	0.7731	0.7726	**0.7732**	0.7614	0.7726	0.7721	0.7722	0.7458
	4	0.8292	0.8292	0.8294	0.8291	**0.8295**	0.8291	0.8291	**0.8295**	0.8291	0.8291	0.8271	0.8264	0.8205
	5	0.8645	0.8647	0.8644	0.863	0.8647	0.8647	0.8645	**0.8649**	0.8647	0.8647	0.8619	0.8611	0.8496
147,091	2	0.7032	0.7035	0.7035	0.7035	0.7035	0.7035	**0.7036**	**0.7036**	0.7035	0.7035	0.7035	0.7036	0.703
	3	0.7816	0.7842	0.783	0.784	0.7842	0.7838	**0.7844**	**0.7844**	0.7842	**0.7844**	0.782	0.782	0.7784
	4	**0.8039**	0.8037	0.8022	0.8031	0.8022	0.7998	0.8028	0.8022	0.7939	0.8021	0.8010	0.8001	0.8014
	5	0.8087	0.8088	0.8091	0.8089	0.8094	0.8109	0.8121	0.8083	0.807	0.8107	0.8107	0.8117	0.808
101,087	2	**0.7295**	**0.7295**	**0.7295**	**0.7295**	**0.7295**	**0.7295**	**0.7295**	**0.7295**	0.7291	0.7294	**0.7295**	0.7291	0.7263
	3	**0.7639**	0.7639	0.7636	0.7637	0.7638	**0.7639**	**0.7639**	**0.7639**	0.7633	0.7638	0.7633	0.7634	0.7622
	4	0.8141	0.8138	0.8089	0.8138	0.8089	0.8035	**0.8142**	0.814	0.8015	0.8091	0.7935	0.8107	0.7834
	5	0.8105	0.8119	0.8101	0.813	0.8154	0.806	0.8153	0.8151	**0.8181**	0.8134	0.8118	0.8068	0.8038
253,027	2	0.6937	0.6935	0.6935	0.6935	0.6935	0.6945	0.6935	**0.6947**	0.6935	0.6935	0.6935	0.6939	0.6915
	3	0.7338	0.7338	0.7326	0.7327	**0.7346**	0.7341	0.7336	0.7341	0.7305	0.7341	0.7324	0.7324	0.7269
	4	0.753	0.7539	0.753	0.7537	**0.7543**	0.754	0.7537	**0.7543**	0.7541	0.7536	0.7505	0.75	0.7477
	5	0.7821	0.7812	0.7961	0.7901	0.7971	0.7989	0.7975	**0.8015**	0.7918	0.7923	0.779	0.7788	0.7867

Bold values indicate the best results

Table 5 SSIM values of benchmark images for 13 different multilevel thresholding methods

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	0.4939	0.4947	0.4947	**0.4949**	0.4944	0.4942	**0.4949**	**0.4949**	0.4944	0.4942	0.4937	0.4945	0.4917
	3	0.5983	0.5983	0.5983	0.5984	0.5978	0.5983	0.5983	**0.5987**	0.5979	0.5978	0.5983	0.5983	0.5934
	4	0.6592	0.6594	0.6589	0.6587	0.6593	0.6526	0.6564	**0.6595**	0.6352	0.6593	0.6585	0.6551	0.6362
	5	0.6773	0.6766	0.6748	0.6708	0.6692	0.6699	0.6695	0.6692	0.6775	0.6706	0.6689	**0.6777**	0.6659
Pepper	2	**0.5897**	**0.5897**	**0.5897**	0.5877	0.5879	**0.5897**	**0.5897**	**0.5897**	0.5844	0.5802	**0.5897**	0.5812	0.5813
	3	**0.6697**	0.6686	0.6691	0.6689	0.6686	0.669	**0.6697**	**0.6697**	0.6686	0.6695	0.6696	0.6687	0.662
	4	0.7212	0.7225	0.7225	0.7216	0.7308	0.7252	0.6894	0.685	0.7127	0.7149	0.7172	0.7215	0.7074
	5	0.7467	0.7473	0.7473	0.7413	0.7462	0.7471	**0.7491**	0.7474	0.7475	0.7473	0.7474	0.7431	0.7407
Goldhill	2	0.1609	**0.1615**	**0.1615**	**0.1615**	0.1554	0.1426	**0.1615**	**0.1615**	**0.1615**	0.1607	**0.1615**	0.1571	0.121
	3	0.2914	0.296	0.2911	0.296	0.2897	0.2946	0.2932	**0.2987**	0.296	0.2977	0.2841	0.2878	0.2809
	4	0.2615	0.2948	0.2716	0.2262	0.2703	0.2752	0.2975	**0.3064**	0.2481	0.2998	0.2394	0.2821	0.2524
	5	0.3376	0.2969	0.322	0.2867	0.3275	0.3386	0.332	**0.3468**	0.3358	0.3249	0.3403	0.335	0.3218
Lena	2	0.5257	0.5261	0.5261	0.5261	**0.5277**	0.5261	0.5261	0.5261	0.5261	0.5275	0.5233	0.5262	0.5257
	3	0.5111	0.5249	0.5244	0.5257	0.5257	0.5095	0.5212	**0.5261**	0.5192	0.5237	0.5168	0.5176	0.5225
	4	0.6186	0.62	0.6189	0.6202	0.6208	0.6141	0.6211	**0.6211**	0.6123	0.6125	0.6138	0.6023	0.6055
	5	0.6329	0.6594	0.6531	0.6621	0.6809	0.6644	**0.6826**	0.6822	0.6642	0.6605	0.6584	0.6312	0.6308
House	2	0.5016	0.5016	0.5016	0.5016	0.5016	**0.5032**	**0.5032**	**0.5032**	0.5032	0.503	0.5011	0.5016	0.4996
	3	0.6446	0.6447	0.6437	0.6452	**0.6466**	0.6402	0.6457	0.6457	0.6185	0.6457	0.6431	0.6341	0.6033
	4	0.6723	0.6739	0.6706	0.6731	0.6649	0.6665	**0.6746**	**0.6746**	0.6715	0.6669	0.6675	0.663	0.6617
	5	0.7289	0.721	0.7225	0.7291	0.7221	0.721	**0.7292**	**0.7292**	**0.7292**	0.7205	0.7215	0.7193	0.7139
12,003	2	**0.3971**	**0.3971**	**0.3971**	**0.3971**	**0.3971**	**0.3971**	**0.3971**	**0.3971**	**0.3971**	0.3956	0.3953	0.3948	0.3875
	3	0.5132	0.513	0.513	0.5124	**0.5145**	0.5141	0.5124	0.5124	0.5124	0.5144	0.5111	0.5124	0.484
	4	0.5712	0.5717	0.57	0.571	0.5726	**0.5732**	0.571	0.571	0.5699	0.5723	0.572	0.5701	0.557
	5	0.6446	0.6458	0.6434	0.6437	**0.6465**	0.6453	0.6446	0.6433	0.6397	0.6463	0.6429	0.6334	0.6281
181,079	2	**0.2326**	**0.2326**	**0.2326**	0.2243	**0.2326**	**0.2326**	**0.2326**	**0.2326**	0.2321	0.2322	0.2267	0.2263	0.2111
	3	0.4075	0.4075	0.4056	0.405	0.4071	0.4067	0.4075	0.4075	0.4019	**0.4096**	0.4077	0.4067	0.3928
	4	**0.6234**	0.6219	0.6231	**0.6234**	**0.6234**	**0.6234**	**0.6234**	**0.6234**	0.6125	0.6181	0.6159	0.6105	0.5783
	5	0.6811	**0.6867**	0.6824	0.6832	0.6845	0.6862	0.6844	0.6862	0.6804	0.684	0.6724	0.6793	0.6625
175,043	2	0.4986	0.4993	0.4993	0.4993	**0.5013**	0.5008	**0.5013**	**0.5013**	0.4925	0.4993	0.4993	0.4986	0.486
	3	0.5905	0.5899	0.59	0.5904	**0.5929**	0.5909	0.5909	0.5909	0.5784	0.5925	0.5806	0.5844	0.5739
	4	0.6714	0.6724	0.6719	0.6734	0.6718	0.6718	0.6742	**0.6755**	0.6751	0.6716	0.6751	0.6715	0.6588
	5	0.7557	0.7553	0.7543	0.7522	0.7511	0.7547	0.7565	0.7561	**0.7567**	0.7494	0.7421	0.7484	0.7433
101,085	2	0.352	**0.3522**	**0.3522**	**0.3522**	0.3519	0.3511	**0.3522**	**0.3522**	**0.3522**	0.3516	0.3514	0.3512	0.34
	3	0.6208	0.6213	0.6214	0.6215	0.6212	0.6212	**0.6218**	**0.6218**	0.5887	0.6212	0.6207	0.6212	0.5477
	4	0.6908	0.6904	0.6901	0.6901	0.6901	0.6906	**0.691**	**0.691**	**0.691**	0.6907	0.6877	0.6873	0.6744
	5	0.7402	0.74	0.7403	0.7372	0.7399	0.74	0.7397	0.7403	0.7399	**0.7409**	0.7393	0.7345	0.7187
147,091	2	0.4923	**0.4928**	**0.4928**	**0.4928**	0.4925	0.4927	**0.4928**	**0.4928**	**0.4928**	0.4927	**0.4928**	**0.4928**	0.4902
	3	0.6053	0.6077	0.6063	0.6075	0.6066	0.6073	**0.6077**	**0.6077**	**0.6077**	0.6053	0.6059	0.6054	0.6022
	4	0.6486	**0.649**	0.6465	0.6479	0.6464	0.6405	0.6476	0.6465	0.625	0.6467	0.6475	0.6474	0.6418
	5	0.6469	0.6482	0.6469	0.6486	0.6493	0.6483	0.6416	**0.6497**	0.644	0.6421	0.6409	0.6403	0.6405
101,087	2	**0.643**	**0.643**	**0.643**	**0.643**	0.642	0.642	**0.643**	**0.643**	0.6422	0.6417	0.6429	**0.643**	0.6403
	3	**0.7055**	**0.7055**	0.7051	0.7052	0.7049	0.7051	**0.7055**	**0.7055**	**0.7055**	0.7051	0.7046	0.7046	0.7056
	4	0.7724	0.7722	0.7675	0.7722	0.772	0.7635	0.7676	**0.7725**	0.7628	0.7674	0.7531	0.765	0.7497
	5	0.7921	0.7956	0.7933	0.7958	0.7983	0.7913	**0.7991**	0.7987	0.7991	0.7964	0.7947	0.792	0.7883
253,027	2	0.4646	**0.4653**	**0.4653**	**0.4653**	**0.4653**	**0.4653**	**0.4653**	**0.4653**	**0.4653**	0.4611	0.464	0.464	0.4633
	3	0.539	0.5388	0.5349	0.5353	0.5411	**0.5421**	0.5379	0.5393	0.5288	0.5393	0.535	0.5339	0.5303
	4	0.5595	0.5636	0.5598	0.5617	**0.5651**	0.565	0.5618	0.5644	0.5644	0.5631	0.5611	0.556	0.5585
	5	0.6357	0.6329	0.6821	0.6641	0.6847	0.6899	0.685	**0.698**	0.6718	0.6673	0.6339	0.6261	0.6622

Bold values indicate the best results

where t_i is the ith threshold value. For this problem, the boundaries of the search space are $l = 0$ and $u = 2^K - 1$, where K is the number of bits indicating one pixel.

2.3 Entropy-based objective function

This criterion is based on maximization of the overall entropy of the segmented image. The entropy of a histogram shows the intra-class compactness and inter-class separability [39]. The remarkable performance of Kapur's method was studied by the researchers [19, 22, 23] and compared with other techniques.

Suppose that L is the number of grey levels in the image *I {1,2,…,L}, N* is the total number of pixels, and $h(i)$ is the number of pixels with grey level i. Then, the probability of occurrence of grey level i in the image I is calculated as follows:

$$p(i) = h(i)/N \tag{36}$$

The objective function f to choose D threshold values is defined using the following equations:

$$\begin{aligned}
&f([t_1, t_2, \ldots, t_m] = H_0 + H_1 + \ldots + H_m \\
&\omega_0 = \sum_{i=0}^{t_1-1} p_i, \quad H_0 = -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0} \\
&\omega_1 = \sum_{i=t_1}^{t_2-1} p_i, \quad H_1 = -\sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1} \\
&\omega_2 = \sum_{i=t_2+1}^{t_3} p_i, \quad H_2 = -\sum_{i=t_2+1}^{t_3} \frac{p_i}{\omega_2} \ln \frac{p_i}{\omega_2} \\
&\omega_m = \sum_{i=t_m+1}^{L} p_i, \; H_m = -\sum_{i=t_m+1}^{L} \frac{p_i}{\omega_m} \ln \frac{p_i}{\omega_m}
\end{aligned} \tag{37}$$

The purpose of P-metaheuristic algorithms used in this paper is to find the proper values of thresholds to maximize this objective function.

3 Experimental results

In this section, the experimental results are presented on several benchmark images. Five popular benchmark images named "Boats", "Pepper", "Goldhill", "Lena", and "House" were used for evaluating the experiments. Moreover, to conduct a better comparison, seven well-known images in the literature were also selected from the Berkeley Segmentation Data Set and Benchmark [63]. The number of selected images is 12,003, 181,079, 175,043, 101,085, 147,091, 101,087, and 253,027.

The original benchmark images and their histograms are shown in Fig. 12. The segmentation process of most of the used images is a difficult task because they have a multimodal histogram. Images such as Pepper and Lena have multiple peaks and valleys, while Goldhill image indicates abruptly changing pixel levels. Other images such as 175,043 and 101,085 show a more smooth distribution compared with other images such as Boats and Pepper.

For a more comprehensive comparison, the eight recently developed P-metaheuristic algorithm were compared with five others. Therefore, it can be said that 13 P-metaheuristic algorithms were compared with each other. Because of the randomness characteristics of P-metaheuristic algorithms, each algorithm is repeated for 50 times and the average results are reported.

The population size and the number of iterations for all algorithms are considered as 30 and 200, respectively. Finding a proper value for parameters is a trial and error task and needs enormous experiments. Hence, in this study, we used the default values for each parameter. Other parameter settings are listed in Table 1.

3.1 Objective function measure

As mentioned previously, an entropy-based criterion is used as the objective function. Table 2 shows the obtained objective function values for each P-metaheuristic algorithm. K is the number of thresholds for each image. In this section, the experimental results for K = 2, 3, 4, and 5 are presented. The best results for each row were boldfaced. As can be observed, the CS algorithm yielded better results than other compared algorithms because it obtained the best results in 37 out of 48 cases. In addition, ICA provided the second rank because it provided the best performance in 31 out of 48 cases. Moreover, TLBO algorithm ranked the third by reaching the best results in 28 of 48 cases.

According to the table, most of the P-metaheuristic algorithms could obtain the same optimal value when K = 2, while the performance is different when K is bigger than 2.

3.2 Peak signal to noise ratio

Peak signal to noise ratio (PSNR) is a popular indicator to measure the quality of segmented images in decibels (DB). PSNR is calculated as follows:

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \tag{38}$$

where $RMSE$ is the root mean square error, which is formulated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j) - \tilde{I}(i,j)\right)^2}{MN}} \tag{39}$$

Table 6 Standard deviation values of benchmark images for 13 different multilevel thresholding methods

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	2.91E−03	2.48E−04	2.48E−04	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	6.34E−03	**1.87E−15**	5.24E−04	4.65E−04	9.93E−03
	3	2.57E−03	8.47E−04	2.16E−03	3.17E−06	2.38E−06	4.19E−04	0.00E+00	**0.00E+00**	3.58E−03	0.00E+00	3.15E−03	3.32E−03	4.11E−02
	4	1.03E−02	2.21E−03	1.28E−03	2.40E−03	1.01E−03	3.28E−02	2.23E−02	**7.72E−05**	3.51E−02	8.46E−04	1.82E−02	9.76E−03	7.30E−02
	5	1.93E−02	2.82E−02	1.28E−02	9.31E−03	4.72E−03	3.40E−03	**1.26E−04**	3.92E−04	7.51E−02	3.10E−03	2.84E−02	1.98E−02	9.23E−02
Pepper	2	**1.87E−15**	**1.87E−15**	2.60E−05	3.47E−05	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	8.69E−04	**1.87E−15**	3.47E−05	3.49E−04	8.54E−03
	3	1.63E−04	**1.87E−15**	3.37E−04	1.46E−04	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	1.50E−03	3.63E−03	4.50E−02
	4	1.82E−03	5.94E−03	2.01E−03	2.92E−03	**1.26E−04**	2.70E−03	7.70E−04	1.95E−04	6.33E−02	3.06E−03	8.29E−03	7.74E−03	5.40E−02
	5	1.07E−02	9.41E−04	1.79E−02	1.81E−02	9.82E−03	1.49E−02	3.74E−4	**3.72E−04**	6.55E−02	1.89E−02	1.84E−02	1.27E−02	5.84E−02
Goldhill	2	5.42E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.65E−01	9.36E−16	**0.00E+00**	5.35E−01	9.36E−16	**0.00E+00**	5.14E−02	4.16E−01
	3	4.90E−01	1.87E−15	3.47E−02	1.87E−15	1.87E−15	7.45E−02	6.54E−04	1.87E−15	6.66E−01	0.00E+00	1.92E−02	5.11E−02	2.51E−01
	4	1.58E−01	1.62E−01	1.08E−01	1.02E−01	1.72E−01	1.93E−01	**1.87E−15**	9.07E−02	4.62E−01	1.87E−15	1.06E−01	1.36E−01	2.87E−01
	5	4.35E−01	3.09E−01	8.09E−02	1.42E−01	1.28E−01	2.01E−01	**2.69E−03**	8.74E−02	5.78E−01	6.46E−03	1.95E−01	2.16E−01	1.31E−01
Lena	2	1.96E−04	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	1.85E−04	7.53E−04	4.11E−02
	3	2.20E−02	1.66E−03	1.91E−03	5.80E−04	**1.87E−15**	3.70E−02	3.74E−15	**1.87E−15**	1.17E−01	3.74E−15	1.48E−02	3.12E−02	1.17E−01
	4	3.24E−02	4.30E−04	6.31E−03	1.14E−02	1.65E−05	7.00E−02	**0.00E+00**	8.77E−05	2.71E−01	7.73E−03	2.97E−02	2.25E−02	1.21E−01
	5	2.67E−01	1.27E−02	2.14E−02	2.02E−02	6.02E−03	9.50E−02	**0.00E+00**	7.63E−04	2.42E−01	1.05E−03	5.78E−02	3.58E−02	2.00E−01
House	2	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	6.34E−04	**1.87E−15**	1.18E−02
	3	7.20E−04	2.29E−03	9.43E−04	1.55E−03	**0.00E+00**	9.91E−03	**0.00E+00**	**0.00E+00**	2.21E−02	3.25E−04	3.13E−03	1.33E−02	2.68E−02
	4	2.82E−03	1.58E−02	4.51E−03	4.24E−03	3.40E−03	6.50E−03	**0.00E+00**	**0.00E+00**	7.82E−03	4.61E−03	8.52E−03	8.95E−03	8.28E−02
	5	9.39E−03	2.07E−02	9.54E−03	5.96E−03	4.04E−03	0.00E+00	**0.00E+00**	1.87E−04	**0.00E+00**	1.73E−03	3.22E−02	2.08E−02	6.84E−02
12,003	2	0.00E+00	0.00E+00	0.00E+00	2.17E−05	0.00E+00	0.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.17E−05	2.89E−05	1.40E−04	6.04E−03
	3	2.08E−04	3.99E−04	2.38E−04	2.54E−04	0.00E+00	0.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.71E−03	2.09E−03	2.13E−02
	4	4.07E−04	1.47E−03	5.84E−04	1.09E−03	2.58E−04	0.00E+00	**0.00E+00**	**0.00E+00**	3.94E−04	1.45E−04	8.62E−03	8.75E−03	3.81E−02
	5	1.91E−03	8.87E−03	4.44E−03	5.96E−03	9.44E−04	**9.07E−05**	1.87E−04	2.36E−04	4.59E−03	6.72E−04	1.84E−02	2.00E−02	5.70E−02
181,079	2	1.87E−15	1.87E−15	1.87E−15	9.53E−05	7.15E−05	**1.87E−15**	**1.87E−15**	**1.87E−15**	1.09E−04	1.87E−15	1.19E−04	4.01E−04	9.98E−03
	3	9.43E−05	3.44E−04	8.47E−05	1.54E−04	**3.74E−15**	**3.74E−15**	**3.74E−15**	**3.74E−15**	1.52E−02	3.74E−15	3.14E−03	2.79E−03	3.32E−02
	4	7.25E−04	6.28E−03	3.53E−03	3.34E−03	**3.74E−15**	2.20E−04	**3.74E−15**	**3.74E−15**	1.09E−01	9.22E−05	1.34E−02	5.87E−03	7.26E−02
	5	7.69E−03	1.49E−02	3.28E−03	1.00E−02	9.26E−04	9.00E−04	9.67E−04	**6.09E−04**	3.46E−02	1.48E−03	3.11E−02	1.74E−02	7.55E−02
175,043	2	2.74E−04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.26E−03	**0.00E+00**	**0.00E+00**	1.91E−04	7.20E−03
	3	1.26E−03	1.73E−03	1.32E−03	2.06E−03	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	1.19E−02	6.45E−05	3.37E−03	5.13E−03	1.21E−02
	4	3.37E−03	2.96E−03	2.28E−03	1.66E−03	1.32E−05	2.15E−05	2.02E−05	**3.74E−15**	1.95E−02	1.70E−04	9.80E−03	1.48E−02	8.99E−02
	5	3.90E−03	1.39E−02	3.09E−03	1.65E−02	4.24E−03	**6.00E−05**	9.21E−04	1.07E−03	1.45E−02	1.14E−03	1.23E−02	7.87E−03	8.03E−02
101,085	2	1.06E−04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.98E−06	2.26E−04	1.08E−02
	3	4.79E−04	7.73E−05	1.70E−04	8.18E−05	**3.74E−15**	**3.74E−15**	**3.74E−15**	**3.74E−15**	8.53E−03	7.73E−05	1.02E−03	7.45E−03	2.00E−02
	4	5.58E−04	2.36E−03	3.30E−04	9.59E−04	3.86E−05	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.62E−05	9.17E−03	1.22E−02	5.28E−02
	5	1.75E−03	1.57E−04	2.58E−03	4.80E−03	1.44E−03	1.34E−05	3.54E−04	2.39E−04	**3.74E−15**	5.28E−04	2.45E−02	1.82E−02	8.03E−02

Table 6 (continued)

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
147,091	2	5.54E−04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.82E−04	6.55E−05	1.11E−02
	3	2.93E−03	**3.74E−15**	3.56E−04	9.16E−05	**3.74E−15**	**3.74E−15**	**3.74E−15**	**3.74E−15**	**3.74E−15**	**3.74E−15**	4.40E−03	4.29E−03	8.26E−02
	4	6.94E−03	5.82E−03	7.36E−03	1.24E−03	3.90E−04	1.77E−02	3.29E−04	**3.74E−15**	4.42E−02	1.34E−02	1.98E−02	1.27E−02	7.30E−02
	5	3.85E−03	2.68E−02	3.30E−03	6.16E−03	4.06E−03	2.85E−03	3.36E−03	**7.02E−04**	3.15E−02	2.53E−02	3.98E−02	1.16E−02	8.83E−02
101,087	2	**1.87E−15**	**1.87E−15**	**1.87E−15**	5.20E−04	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	1.73E−03	**1.87E−15**	5.79E−04	5.78E−04	1.47E−02
	3	1.13E−03	**1.87E−15**	1.38E−03	9.17E−04	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	4.01E−03	**1.87E−15**	4.92E−03	6.04E−03	7.07E−02
	4	2.19E−03	**3.74E−15**	1.88E−02	4.34E−03	2.45E−04	4.65E−02	1.77E−02	2.15E−04	5.69E−02	1.76E−02	3.05E−02	9.63E−03	7.82E−02
	5	2.61E−02	4.50E−02	2.26E−02	1.08E−02	3.89E−03	6.78E−02	**3.74E−15**	3.23E−04	2.01E−01	7.50E−03	3.13E−02	3.61E−02	9.63E−02
253,027	2	5.83E−05	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	**1.87E−15**	2.62E−05	1.07E−04	3.05E−02
	3	9.00E−05	1.38E−03	3.27E−04	3.22E−04	1.87E−15	6.75E−05	1.71E−04	**1.87E−15**	1.63E−02	1.68E−04	2.60E−03	2.62E−03	4.97E−02
	4	1.54E−03	3.77E−03	1.27E−03	1.63E−03	2.83E−04	4.07E−04	4.07E−04	**0.00E+00**	7.51E−02	4.44E−04	1.08E−02	1.02E−02	3.35E−02
	5	2.91E−03	1.49E−02	6.80E−03	5.71E−03	3.14E−03	4.85E−03	2.43E−04	**7.86E−05**	1.87E−02	4.24E−03	1.91E−02	1.47E−02	5.64E−02

Bold values indicate the best results

Here, M and N are the size of image, and I and $\tilde{I}$ are the original and segmented images. A higher value of PSNR shows a better quality of the segmented image. Table 3 shows PSNR values of benchmark images. According to this table, in most of the cases (28 out of 48) the CS algorithm has presented higher PSNR compared with others. ICA and TLBO have been ranked second and third, respectively. ICA and TLBO provided the best PSNR in 12 and 8 out of 48, respectively. In addition, it is evident that as the number of threshold values increases, PSNR often raises.

3.3 Feature similarity index

Feature similarity index (FSIM) [70] is another image quality indicator based on human visual system. FSIM works on two low-level features, namely high phase congruency (PC), and gradient magnitude (GM). PC is a dimensionless measure that shows the importance of local structure. GM is used to encode contrast information.

FSIM is formulated as:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(X) PC_m(X)}{\sum_{x\in\Omega} PC_m(X)} \tag{40}$$

where Ω shows the whole image, $S_L(X)$ is the similarity between the original image and its segmented image and PC is the phase congruence. $S_L(X)$ and PC is calculated as follows:

$$S_L(X) = S_{PC}(X) S_G(X) \tag{41}$$

$$S_{PC}(X) = \frac{2PC_1(X)PC_2(X) + T_1}{PC_1^2(X) + PC_2^2(X) + T_1} \tag{42}$$

$$S_G(X) = \frac{2G_1(X)G_2(X) + T_2}{G_1^2(X) + G_2^2(X) + T_2} \tag{43}$$

where T_1 and T_2 are constants. G is the gradient descent of an image and is calculated as:

$$G = \sqrt{G_x^2 + G_y^2} \tag{44}$$

In addition, PC is defined as:

$$PC(X) = \frac{E(X)}{(\varepsilon + \sum_n A_n(X))} \tag{45}$$

where $E(X)$ is the magnitude of the response vector at position X on scale n, ε is a small number, and $A_n(X)$ is the local amplitude on scale n.

A higher value of FSIM indicates better performance. Table 4 shows FSIM values for 13 different multilevel

Table 7 The obtained objective function values for benchmark images on higher dimensions

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	10	33.1463	33.1701	33.0298	33.0546	32.9282	33.2193	**33.2747**	33.2282	33.0819	33.2024	32.7894	32.8483	32.0237
	15	42.5963	42.692	41.9275	42.051	41.4383	42.4704	**43.0047**	42.7775	42.8005	42.1332	41.1504	42.1609	41.6089
	20	50.0062	49.4478	48.9813	49.6707	48.2474	49.8903	**51.1721**	50.6592	50.6994	49.8974	48.1668	49.9512	49.7044
Pepper	10	32.4573	32.6867	32.4971	32.4892	32.2858	32.5956	**32.8332**	32.7677	32.5787	32.5133	32.0542	32.2631	31.4526
	15	40.8905	41.4987	40.7788	41.0816	40.6041	41.7888	**41.9027**	41.6022	41.4516	41.0015	39.7267	41.0887	40.5284
	20	47.669	49.0154	47.1258	47.6536	46.8483	48.8269	**49.3608**	48.6576	49.007	47.9482	46.4552	48.0988	47.8892
Goldhill	10	18.4177	19.181	18.454	20.6538	18.5473	17.6067	18.6253	**20.6924**	19.076	18.2365	18.2868	19.9387	20.0254
	15	21.7338	22.7196	21.8484	**25.0241**	21.4852	20.9857	22.6214	24.7088	23.2675	20.1261	21.1768	24.5314	22.1501
	20	23.7199	25.0015	23.8727	28.0254	23.3017	23.1403	27.6235	28.1843	**28.1897**	23.3001	23.3187	28.048	23.5656
Lena	10	32.4233	32.5501	32.4097	32.5042	32.3434	32.1785	32.6125	**32.7109**	32.2534	31.3641	32.1307	32.2811	31.5655
	15	41.2158	41.6769	41.0198	41.4486	40.872	40.5125	41.2112	**41.9929**	41.9754	39.8718	40.375	41.3811	40.9437
	20	48.4736	49.1377	47.8243	48.6508	47.2046	48.0599	48.8555	49.591	**49.8185**	46.3987	47.247	48.9604	48.8612
12,003	10	33.4755	33.5005	33.3511	33.3272	33.2829	33.5278	**33.5673**	33.5394	33.4551	33.5087	32.9363	33.1404	32.3587
	15	42.5588	42.6855	42.0943	42.3011	41.7948	42.7912	**42.9587**	42.765	42.8929	42.5711	41.3947	42.2545	41.6328
	20	50.1554	50.1473	48.9419	49.7137	48.4768	49.901	**51.0899**	50.4945	50.8209	49.7871	48.1647	49.8493	49.6731
House	10	32.0331	32.0038	31.9733	31.8465	31.8154	32.0209	**32.1124**	32.0832	32.0381	32.0041	31.5125	31.709	30.8474
	15	40.3614	40.4989	40.1001	40.4068	39.9362	40.9806	**41.2119**	40.9865	40.6838	40.5957	39.3341	40.471	39.9014
	20	47.5528	48.0992	46.2559	47.3282	46.2534	47.8314	**48.7051**	47.9592	48.0948	47.3666	45.2928	47.4477	47.4533
181,079	10	32.8129	32.7094	32.6443	32.7981	32.5576	32.5853	32.5896	**32.9782**	32.7642	32.498	32.1597	32.5436	31.7525
	15	42.4127	42.264	41.6358	42.0917	41.3702	41.813	42.1501	**42.6464**	42.6342	41.5367	41.0067	42.0568	41.5433
	20	49.6058	50.3827	48.9025	49.7024	48.2358	49.2934	50.245	50.5671	**50.8919**	48.6872	47.897	49.9691	49.7248
175,043	10	33.2289	33.2219	33.0105	33.0755	32.9479	33.2014	33.2109	**33.3123**	33.2399	33.1246	32.7386	32.8584	32.1474
	15	42.4848	42.5803	42.0298	42.4035	41.5406	42.4196	42.6461	**42.861**	42.6709	42.1154	41.192	42.349	41.7525
	20	49.9842	50.1495	48.7859	49.794	48.1448	49.6046	50.4624	50.6062	**50.7185**	49.0552	48.1321	49.9161	49.7738
101,085	10	33.7581	33.7276	33.6822	33.5756	33.5208	33.8032	33.8085	33.7807	**33.8177**	33.735	33.3384	33.4087	32.6261
	15	43.2378	43.1802	42.6335	42.8129	42.3949	43.2078	**43.4499**	43.3102	43.433	42.9395	42.0022	42.8069	42.2175
	20	50.8945	50.8034	49.5715	50.31	49.3846	50.3914	**51.655**	51.1674	51.4657	50.5041	48.7748	50.5235	50.2322
147,091	10	33.6944	33.7056	33.6135	33.5591	33.5002	33.809	**33.8138**	33.7998	33.7218	33.7279	33.3511	33.379	32.5643
	15	43.2349	43.3094	42.6426	42.7502	42.3736	43.1349	**43.576**	43.3384	43.3236	42.5828	42.1191	42.7631	42.2081
	20	50.7607	50.5142	49.5325	50.1458	49.1468	50.1785	**51.7141**	51.1368	51.4599	50.3067	48.7736	50.5651	50.3208
101,087	10	32.2712	32.2052	32.2038	32.0705	32.0947	32.2885	**32.3215**	32.295	32.2458	32.2208	31.7511	31.8707	31.0526
	15	41.9419	41.7468	41.4418	41.5375	41.0606	41.6822	**42.1866**	42.0138	42.0913	41.6643	40.6739	41.5168	40.8586
	20	49.8104	49.7923	48.5064	49.1744	48.2288	49.2417	**50.5205**	49.9767	50.2875	49.4809	47.7229	49.248	49.0639
253,027	10	32.4358	32.4727	32.338	32.3373	32.2734	32.5244	**32.5413**	32.4929	32.3901	32.4368	31.8726	32.079	31.3085
	15	41.3069	41.5729	40.936	41.2149	40.7302	41.6847	**41.9655**	41.6819	41.8276	41.3734	39.977	41.2268	40.5955
	20	48.7236	48.5875	47.2668	48.3249	47.1994	48.6714	**49.7564**	49.2621	49.3127	48.1903	46.4469	48.6303	48.6023

Bold values indicate the best results

thresholding methods. It indicates a relatively high correlation with the results of maximum entropy in Table 1. Similar to Table 1, the CS algorithm presents the competitive results compared with others according to the FSIM measure. The CS algorithm obtained the best performance in 32 out of 48 cases, while ICA (second rank) provided the best results in 21 out of 48 cases. TLBO and GSA achieved the third rank simultaneously because both of them had the best performance in 9 out of 48 cases.

3.4 Structural similarity index

Structural similarity index measure (SSIM) [71] is another quality assessment measure that is used to evaluate the performance of the algorithms. SSIM is based on the degradation of structural information. It combines luminance, contrast, and structure to yield a quality measure. Moreover, SSIM satisfies symmetry, boundedness, and unique maximum conditions. SSIM is mathematically represented as:

Table 8 Friedman rank for benchmark images

K	Images	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
2	Boats	11	7.5	7.5	3.5	3.5	3.5	3.5	3.5	12	3.5	10	9	13
	Pepper	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	12	5.5	5.5	11	13
	Goldhill	11	3.5	3.5	3.5	3.5	9	8	3.5	12	13	3.5	7	10
	Lena	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	11	12	13
	House	6	6	6	6	6	6	6	6	6	6	12	6	13
	12,003	6	6	6	6	6	6	6	6	6	6	6	12	13
	181,079	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	11	12	13
	175,043	10.5	5	5	5	5	5	5	5	12	5	5	10.5	13
	101,085	6	6	6	6	6	6	6	6	6	6	6	12	13
	147,091	12	5	5	5	5	5	5	5	5	5	11	10	13
	101,087	4.5	4.5	4.5	9	4.5	4.5	4.5	4.5	12	4.5	11	10	13
	203,027	6	6	6	6	6	6	6	6	6	6	6	12	13
	Average rank	7.46	5.5	5.5	5.54	5.17	5.63	5.54	5.17	8.33	5.96	8.17	10.29	12.75
	Overall rank	9	3.5	3.5	5.5	1.5	7	5.5	1.5	11	8	10	12	13
3	Boats	9	7	8	3	3	6	3	3	11	3	12	10	13
	Pepper	8.5	4	10	8.5	4	4	4	4	4	4	11	12	13
	Goldhill	9	2.5	6	2.5	2.5	8	10	2.5	11	13	5	7	12
	Lena	7	4	5	3	1.5	10	12.5	1.5	9	12.5	6	8	11
	House	5.5	7	8	5.5	2	10	2	2	12	4	9	11	13
	12,003	7.5	7.5	9.5	9.5	3.5	3.5	3.5	3.5	3.5	3.5	11	12	13
	181,079	3.5	9	7.5	7.5	3.5	3.5	3.5	3.5	12	3.5	10	11	13
	175,043	8.5	6	7	8.5	3	3	3	3	12	3	10	11	13
	101,085	8.5	3.5	8.5	7	3.5	3.5	3.5	3.5	11	3.5	10	12	13
	147,091	10	4.5	9	4.5	4.5	4.5	4.5	4.5	4.5	4.5	11	12	13
	101,087	8	3.5	9	7	3.5	3.5	3.5	3.5	10	3.5	12	11	13
	203,027	5	9	8	7	2	2	5	2	12	5	10	11	13
	Average rank	7.5	5.63	7.96	6.13	3.04	5.13	4.83	3.04	9.33	5.25	9.75	10.67	12.75
	Overall rank	8	6	9	7	1.5	4	3	1.5	10	5	11	12	13
4	Boats	8	4	5	6	2	10	7	1	12	3	9	11	13
	Pepper	4	8	7	9	3	6	2	1	12	5	10	11	13
	Goldhill	8	4	7	1	3	12	11	2	9	13	5	6	10
	Lena	7	3	5	4	1	10	12	2	9	13	6	8	11
	House	7.5	7.5	10	5	4	6	1.5	1.5	9	3	12	11	13
	12,003	7	8	9	10	4.5	2	2	2	6	4.5	11	12	13
	181,079	6	7	8.5	8.5	2	4.5	2	2	12	4.5	11	10	13
	175,043	7	6	8	9	2.5	2.5	2.5	2.5	10	5	11	12	13
	101,085	7	9	8	10	3.5	3.5	3.5	3.5	3.5	3.5	11	12	13
	147,091	8	5	7	4	3	10	2	1	12	6	11	9	13
	101,087	4	1	8	5	2.5	10	6	2.5	11	7	12	9	13
	203,027	8.5	6	8.5	7	2	3.5	3.5	1	12	5	11	10	13
	Average rank	6.83	5.71	7.58	6.54	2.75	6.67	4.58	1.83	9.79	6.04	10	10.08	12.58
	Overall rank	8	4	9	6	2	7	3	1	10	5	11	12	13

Table 8 (continued)

K	Images	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
5	Boats	6	8	7	13	5	3	1	2	9	4	11	10	12
	Pepper	6	3	7	13	4	5	1	2	11	8	10	9	12
	Goldhill	11	5	8	9	2	13	4	1	12	7	6	3	10
	Lena	9	3	4	11	2	8	12	1	7	13	6	5	10
	House	7	9	8	13	6	2	2	4	2	5	11	10	12
	12,003	6	7	8.5	13	5	1	2.5	2.5	8.5	4	10	11	12
	181,079	7	8	6	13	3.5	3.5	2	1	9	5	11	10	12
	175,043	7	6	8	13	5	1	2	3	9	4	11	10	12
	101,085	8	2	9	13	7	2	4.5	4.5	2	6	11	10	12
	147,091	5	8	6	13	4	3	2	1	9	7	11	10	12
	101,087	7	8	6	1	4	10	2	3	11	5	9	12	13
	203,027	6	7	8	9	5	4	1.5	1.5	10	3	12	11	13
	Average rank	7.08	6.17	7.13	11.17	4.38	4.63	3.04	2.21	8.29	5.92	9.92	9.25	11.83
	Overall rank	7	6	8	12	3	4	2	1	9	5	11	10	13

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{46}$$

where μ_x and μ_y are the mean intensity of images x and y, σ_x^2 and σ_y^2 indicate the variance of images x and y, σ_{xy} is the covariance of x and y, C_1 and C_2 are constants to balance the calculations. A higher value of SSIM indicates better performance. Table 5 shows SSIM value between an original image and its segmented image. The best result for each row is boldfaced. The results of Table 5 is consistent with the previous results. As can be seen, CS algorithm has presented higher SSIM in 30 out of 48 cases, and, therefore CS achieved the first rank, while the second rank went to ICA.

3.5 Stability analysis

In general, the P-metaheuristic algorithms are stochastic algorithms. Hence, the results may not be the same in each run of the algorithm. Therefore, stability analysis is necessary to assess the P-metaheuristic algorithm. This assessment will show which algorithm is more stable. Standard deviation (STD) is used to analyze the stability of the algorithms. It is defined as follows:

$$STD = \sqrt{\sum_{i=1}^{k} \frac{(\sigma_i - \mu)^2}{k}} \tag{47}$$

where k is the number of runs of each algorithm (K = 50), σ_i is the best objective value of the ith run of the algorithm, and μ is the average value of σ. The lower value of STD shows the more stability for the algorithm. The standard deviation for 50 runs of each algorithm is provided in Table 6. The obtained results revealed that CS and ICA algorithms are more stable than other algorithms.

3.6 The search ability on higher dimension problems

In this section, the P-metaheuristic algorithms were compared on each image with K = 10, 15, and 20 to evaluate the performance of algorithms on higher dimensions. Table 7 provides the objective function mean values obtained by each P-metaheuristic algorithm. Consequently, the ICA algorithm provides the best objective function values in most of the cases. ICA obtained the best performance in 23 out of 48 cases, while CS provided the best results in 7 out of 48 cases.

3.7 Statistical analysis

In this section, Friedman test and Wilcoxon signed rank test were applied as a non-parametric statistical test to compare the performance of P-metaheuristic algorithms. Such statistical test is necessary because of stochastic and random searching characteristics of P-metaheuristic algorithms. The objective of Friedman test is to find the significant differences among the behavior of P-metaheuristic algorithms. The null hypothesis in Friedman test, H_0, states the equality of medians among algorithms, whereas the alternative hypothesis, H_1, shows the difference. Significance level (α) illustrates the probability of the rejection of null hypothesis while it is true. If P-value were less than significance level, H_0 would be rejected. More details about Friedman test can be found in [72].

Table 9 Friedman rank for benchmark images on higher dimensions

k	Images	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
10	Boats	6	5	9	8	10	3	1	2	7	4	12	11	13
	Pepper	9	3	7	8	10	4	1	2	5	6	12	11	13
	Goldhill	10	5	9	2	8	13	7	1	6	12	11	4	3
	Lena	5	3	6	4	7	10	2	1	9	13	11	8	12
	House	4	7	8	9	10	5	1	2	3	6	12	11	13
	12,003	6	5	8	9	10	3	1	2	7	4	12	11	13
	181,079	2	5	6	3	9	8	7	1	4	11	12	10	13
	175,043	3	4	9	8	10	6	5	1	2	7	12	11	13
	101,085	5	7	8	9	10	3	2	4	1	6	12	11	13
	147,091	7	6	8	9	10	2	1	3	5	4	12	11	13
	101,087	4	7	8	10	9	3	1	2	5	6	12	11	13
	203,027	6	4	8	9	10	2	1	3	7	5	12	11	13
	Average rank	5.58	5.08	7.83	7.33	9.42	5.17	2.5	2	5.08	7	11.83	10.08	12.08
	Overall rank	6	3.5	9	8	10	5	2	1	3.5	7	12	11	13
15	Boats	5	4	10	9	12	6	1	3	2	8	13	7	11
	Pepper	9	4	10	7	11	2	1	3	5	8	13	6	12
	Goldhill	9	5	8	1	10	12	6	2	4	13	11	3	7
	Lena	6	3	8	4	10	11	7	1	2	13	12	5	9
	House	9	6	10	8	11	3	1	2	4	5	13	7	12
	12,003	7	5	10	8	11	3	1	4	2	6	13	9	12
	181,079	3	4	9	6	12	8	5	1	2	11	13	7	10
	175,043	5	4	10	7	12	6	3	1	2	9	13	8	11
	101,085	4	6	10	8	11	5	1	3	2	7	13	9	12
	147,091	5	4	9	8	11	6	1	2	3	10	13	7	12
	101,087	4	5	10	8	11	6	1	3	2	7	13	9	12
	203,027	7	5	10	9	11	3	1	4	2	6	13	8	12
	Average rank	6.08	4.58	9.5	6.92	11.08	5.92	2.42	2.42	2.67	8.58	12.75	7.08	11
	Overall rank	6	4	10	7	12	5	1.5	1.5	3	9	13	8	11
20	Boats	4	10	11	9	12	7	1	3	2	6	13	5	8
	Pepper	9	2	11	10	12	4	1	5	3	7	13	6	8
	Goldhill	8	6	7	4	11	13	5	2	1	12	10	3	9
	Lena	8	3	10	7	12	9	6	2	1	13	11	4	5
	House	6	2	11	10	12	5	1	4	3	9	13	8	7
	12,003	4	5	11	9	12	6	1	3	2	8	13	7	10
	181,079	8	3	10	7	12	9	4	2	1	11	13	5	6
	175,043	5	4	11	7	12	9	3	2	1	10	13	6	8
	101,085	4	5	11	9	12	8	1	3	2	7	13	6	10
	147,091	4	6	11	10	12	9	1	3	2	8	13	5	7
	101,087	4	5	11	9	12	8	1	3	2	6	13	7	10
	203,027	4	8	11	9	12	5	1	3	2	10	13	6	7
	Average rank	5.67	4.92	10.5	8.33	11.92	7.67	2.17	2.92	1.83	8.92	12.58	5.67	7.92
	Overall rank	5.5	4	11	9	12	7	2	3	1	10	13	5.5	8

The ranking calculation is the first step in the Friedman test. Average ranking is applied to rank each algorithm and equal values are given to the average ranks. Tables 8 and 9 show the ranks computed for each used algorithm. As can be observed, the TLBO and CS algorithms have provided the first rank for K = 2 and 3. In addition, BA algorithm has obtained the first rank for K = 20. The CS algorithm has presented the first rank in the remaining algorithms. In another word, it can be said that the CS algorithm shows the first rank for all cases except K = 20.

Table 10 Chi-square and P-value for different threshold numbers

K	Chi-square value	P-value
2	87.43	1.55041E-13
3	91.15	2.95385E-15
4	89.35	6.60582E-14
5	86.51	2.33361E-13
10	97.57	1.66260E-15
15	113.26	1.35076E-18
20	117.86	1.65152E-19

Table 11 Pairwise comparison between CS algorithm and other algorithms

Comparison	Better	Worse	Similar
CS versus WOA	39	0	9
CS versus GWO	30	2	16
CS versus COA	37	0	11
CS versus BBO	32	2	14
CS versus TLBO	19	1	28
CS versus GSA	22	4	22
CS versus ICA	15	5	28
CS versus BA	35	2	11
CS versus FF	28	0	20
CS versus PSO	42	0	6
CS versus DE	47	0	1
CS versus GA	48	0	0

Table 12 Wilcoxon signed rank test results

Comparison	P-value
CS versus WOA	5.2553E-08
CS versus GWO	1.5427E-06
CS versus COA	1.1402E-07
CS versus BBO	1.7091E-05
CS versus TLBO	1.0335E-04
CS versus GSA	2.6298E-05
CS versus ICA	0.0012
CS versus BA	1.4597E-07
CS versus FF	3.7623E-06
CS versus PSO	1.6479E-08
CS versus DE	2.3968E-09
CS versus GA	4.4406E-10

The Chi-square (χ^2) values and P-value for different threshold numbers are shown in Table 10. From Chi-square distribution table, the critical value for $(13-1)=12$ degree of freedom with 0.05 significance level is 21.026. According to Table 10, Chi-square values for all threshold numbers are greater than the critical value. It means that H_0 is rejected and H_1 is accepted in all cases. Moreover, the P-value for all threshold numbers is very small, which confirms the rejection of H_0. This shows a significant difference in the behavior of the algorithms.

Given the obtained results, we could say that in general the CS algorithm presented the best results. Table 11 displays the cases, in which the CS algorithm is better, worse, and/or equal to other algorithms. As can be seen in the table, the CS algorithm is performed better than other ones in most of the cases. The Wilcoxon signed rank test is used to compare whether there is a significant difference between the results, statistically and to validate the results. This test compares two algorithms. This test is to answer whether or not there is a significant difference between the algorithms. Therefore, this test is administered between the CS algorithm and other ones. This test reports a P-value as an output, which determines the significance level of the two algorithms. Here, if the P-values are less than 0.05, then there is a significant difference between the two algorithms, statistically. Table 12 illustrates the results of Wilcoxon signed rank test between CS and other algorithms. The table confirms that there is a significant difference between the CS and other algorithms, statistically and solutions are not received by chance because the P-values obtained are less than 0.05 in all cases.

3.8 The effects of other objective functions

To create a more reliable result, another objective function was evaluated based on Cross Entropy. Suppose that $F=\{f_1,f_2,\ldots,f_N\}$ and $G=\{g_1,g_2,\ldots,g_N\}$are two probability distributions on an equal set. The cross entropy between F and G is defined as follows:

$$D(F,G)=\sum_{i=1}^{N} f_i \log\frac{f_i}{g_i} \tag{48}$$

The objective is to reach the minimum of cross entropy between the main image and the segmented one. Suppose that I is the main image, h(i), i = 1,2,…,L is the corresponding histogram, and L is the number of gray levels. The segmented image $I\ _t$ is elucidated using threshold value t as follows:

$$I_t(x,y)=\begin{cases}\mu(1,t) & I(x,y)<t\\ \mu(t,L+1) & I(x,y)\geq t\end{cases} \tag{49}$$

where,

$$\mu(a,b)=\sum_{i=a}^{b-1} ih(i) \Bigg/ \sum_{i=a}^{b-1} h(i) \tag{50}$$

Then, the cross entropy is calculated as follows:

Table 13 The mean objective function value on 50 independent runs concerning cross entropy objective function

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
Boats	2	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8244	639.8103
	3	640.4305	640.4305	640.4305	640.4304	640.4305	640.4305	640.4305	640.4305	640.4305	640.4305	640.4305	640.4295	640.3907
	4	640.7322	640.7308	640.7319	640.7306	640.7321	640.7324	640.7324	640.7324	640.7324	640.7322	640.7324	640.7087	640.6788
	5	640.9251	640.9274	640.9269	640.9249	640.9278	640.928	640.928	640.928	640.928	640.9258	640.928	640.9231	640.8745
Pepper	2	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0414	604.0032
	3	604.6309	604.6306	604.6308	604.6253	604.6309	604.6248	604.6285	604.6309	604.6237	604.6284	604.6309	604.6249	604.5896
	4	605.0741	605.0751	605.0745	605.0742	605.075	605.0751	605.0751	605.0751	605.0751	605.0749	605.0751	605.0751	604.974
	5	605.274	605.2791	605.2785	605.2783	605.2808	605.2712	605.2696	605.2808	605.2651	605.2303	605.2737	605.2782	605.2152
Goldhill	2	538.7864	538.7864	538.7864	538.7864	538.7864	538.7864	538.7864	538.7864	538.7854	538.7864	538.7864	538.7864	538.7416
	3	539.4924	539.4924	539.4924	539.4921	539.4924	539.4924	539.4921	539.4924	539.4924	539.4924	539.4921	539.4924	539.4453
	4	539.8604	539.8612	539.8612	539.8601	539.8612	539.8604	539.8601	539.8612	539.8538	539.8574	539.8612	539.8541	539.787
	5	540.0407	540.0509	540.0508	540.0487	540.0531	540.0502	540.0466	540.0531	540.0351	540.0332	540.0518	540.0518	540.0034
Lena	2	504.8954	504.8954	504.8954	504.8953	504.8954	504.8954	504.8954	504.8954	504.8954	504.8954	504.8954	504.8954	504.8521
	3	505.6631	505.6627		505.6630	505.6633	505.6633	505.6633	505.6633	505.6633	505.6633	505.6633	505.6633	505.5522
	4	505.9565	505.9568	505.9564	505.9552	505.9568	505.9568	505.9568	505.9568	505.9568	505.9567	505.9568	505.9555	505.8916
	5	506.144	506.1443	506.1421	506.1414	506.1443	506.1446	506.1446	506.1446	506.1446	506.1387	506.1446	506.1445	506.0596
House	2	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6968	523.6801
	3	524.7369	524.737	524.7369	524.7369	524.737	524.737	524.737	524.737	524.737	524.737	524.737	524.737	524.6227
	4	525.2139	525.2141	525.2136	525.2120	525.2141	525.2141	525.214	525.2141	525.2141	525.214	525.2141	525.2141	525.1196
	5	525.4683	525.4645	525.467	525.4650	525.4683	525.4685	525.4683	525.4685	525.4685	525.4492	525.4683	525.4581	525.4085
12,003	2	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.4009	598.3784
	3	599.0484	599.0484	599.0484	599.0483	599.0484	599.042	599.0484	599.0484	599.0484	599.0347	599.0484	599.0479	598.9988
	4	599.5234	599.5236	599.5225	599.5225	599.5236	599.5152	599.5236	599.5236	599.5112	599.5079	599.5236	559.5225	599.4724
	5	599.796	599.7958	599.7954	599.7950	599.7972	599.7972	599.7973	599.7973	599.7973	599.7154	599.7973	599.7434	599.7598
181,079	2	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0804	532.0516
	3	532.6512	532.6498	532.6512	532.6509	532.6512	532.6512	532.6512	532.6512	532.6512	532.6512	532.6512	532.6511	532.6044
	4	533.0148	533.0144	533.0147	533.0138	533.015	533.015	533.015	533.015	533.015	533.0149	533.015	532.9383	532.9767
	5	533.1892	533.1889	533.1887	533.1882	533.1894	533.1896	533.1896	533.1896	533.1896	533.1841	533.1895	533.1625	533.1417
175,043	2	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5667	509.5328
	3	510.0821	510.082	510.0821	510.0817	510.0822	510.0822	510.0822	510.0822	510.0822	510.0822	510.0822	510.0822	510.001
	4	510.3459	510.3455	510.3458	510.3433	510.346	510.346	510.346	510.346	510.346	510.346	510.346	510.3425	510.2911
	5	510.497	510.4972	510.4959	510.4936	510.4969	510.4972	510.497	510.4972	510.4972	510.4963	510.4971	510.4970	510.4287
101,085	2	450.1748	450.1746	450.1748	450.1747	450.1748	450.1748	450.1748	450.1748	450.1748	450.1748	450.1748	450.1748	450.1305
	3	451.5201	451.5198	451.5199	451.5195	451.5201	451.5201	451.5201	451.5201	451.5201	451.5201	451.5201	451.5201	451.4422
	4	452.2069	452.2069	452.2057	452.2050	452.207	452.207	452.207	452.2071	452.2071	452.2064	452.2071	452.2026	452.0864
	5	452.5864	452.5865	452.5808	452.5791	452.5866	452.5867	452.5867	452.5868	452.587	452.5241	452.5869	452.5741	452.5184

Table 13 (continued)

Images	K	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
147,091	2	518.5922	518.5922	518.5922	518.5921	518.5922	518.5922	518.5922	518.5922	518.5851	518.5922	518.5922	518.5922	518.5625
	3	519.7165	519.7166	519.7164	519.7159	519.7166	519.7166	519.7166	519.7166	519.7166	519.7166	519.7166	519.7164	519.6323
	4	520.2329	520.2338	520.2322	520.2300	520.2337	520.2335	520.2337	520.2338	520.2338	520.2328	520.2337	520.2338	520.1345
	5	520.5449	520.5469	520.543	520.5436	520.5467	520.5467	520.5467	520.5469	520.5469	520.517	520.5469	520.5442	520.4612
101,087	2	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4374	781.4293
	3	782.2248	782.2243	782.2248	782.2245	782.2248	782.2248	782.2248	782.2248	782.2248	782.2248	782.2248	782.2248	782.1832
	4	782.664	782.6642	782.6635	782.6633	782.6641	782.6642	782.6641	782.6642	782.6641	782.6641	782.6642	782.6597	782.6202
	5	782.8644	782.8646	782.8637	782.8603	782.8644	782.8645	782.8644	782.8647	782.8648	782.8623	782.8648	782.8647	782.8302
253,027	2	534.7507	534.7508	534.7509	534.7509	534.7509	534.7509	534.7509	534.7509	534.7509	534.7509	534.7509	534.7509	534.7318
	3	535.2918	535.2916	535.2918	535.2915	535.2918	535.2918	535.2918	535.2918	535.2918	535.2918	535.2918	535.2916	535.252
	4	535.546	535.5463	535.5462	535.5457	535.5463	535.5464	535.5464	535.5464	535.5464	535.5464	535.5464	535.5461	535.5117
	5	535.6701	535.6842	535.6835	535.6811	535.6844	535.6844	535.6842	535.6843	535.6844	535.6756	535.6844	535.6841	535.6369

$$D(t) = \sum_{i=1}^{L} ih(i)\log(i) - \sum_{i=1}^{t-1} ih(i)\log(\mu(1,t)) - \sum_{i=t}^{L} ih(i)\log(\mu(t,L+1)). \quad (51)$$

The objective is to obtain threshold value t^* by minimizing the cross entropy in accordance with the following equation:

$$t* = \arg\min_{t}\{D(t)\}. \quad (52)$$

Since the first factor for an image is a constant value, the objective function could be revised as follows:

$$\begin{aligned} D(t) &= ih(i)\log(\mu(1,t)) - \sum_{i=t}^{L} ih(i)\log(\mu(t,L+1)) \\ &= -\sum_{i=1}^{t-1} -m^1(1,t)\log\left(\frac{m^1(1,t)}{m^0(1,t)}\right) \\ &\quad - m^1(t,L+1)\times\log\left(\frac{m^1(t,L+1)}{m^0(t,L+1)}\right) \end{aligned} \quad (53)$$

where, $m^0(a,b) = \sum_{i=a}^{b-1} h(i)$ and $m^1(a,b) = \sum_{i=a}^{b-1} ih(i)$ are the momentum 0 and 1 on a minor interval of image histogram, respectively. Suppose that we want to obtain C threshold values for $t_1, t_2, \ldots, t_c$. To make the computation simple, two dummy variables of $t_0 = 1$ and $t_{c+1} = L + 1$ were defined. Hence, the final objective function is described as follows:

$$f(t_1, t_2, \ldots, t_c) = -\sum_{i=1}^{c+1} m^1(t_{i-1}, t_i)\log\left(\frac{m^1(t_{i-1}, t_i)}{m^0(t_{i-1}, t_i)}\right) \quad (54)$$

Here, the objective is to obtain the optimal value for $x = [t_1, t_2, \ldots, t_c]$ vector, such that the Eq. (54) becomes minimize. Table 13 indicates the mean objective function value on 50 independent runs. Moreover, the rank of each row is depicted in Table 14. Given Tables 13 and 14, the COA, TLBO, GSA, ICA, CS, PSO, and DE algorithms could reach to the best value of the objective function for K = 2. As can be seen, several algorithms were reached to the best value of the objective function for K = 2, simultaneously, the reason of which is the small size of the solution space. TLBO and CS algorithms have had the best results for K = 3. However, only the CS algorithm held the first rank for K = 4, 5. The CS algorithm not only achieved the best objective function value in common with other algorithms (K = 2 and 3) at lower dimensions, but also reached exclusively the best rank at higher dimensions (K = 4 and 5), where the solution space is larger. Therefore, we could

Table 14 Friedman rank for benchmark images concerning cross entropy objective function

K	Images	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
2	Boats	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	Pepper	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	Goldhill	6	6	6	6	6	6	6	6	12	6	6	6	13
	Lena	6	6	6	12	6	6	6	6	6	6	6	6	13
	House	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	12,003	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	181,079	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	175,043	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	101,085	5.5	12	5.5	11	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	13
	147,091	5.5	5.5	5.5	11	5.5	5.5	5.5	5.5	12	5.5	5.5	5.5	13
	101,087	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	13
	203,027	12	11	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	13
	Average rank	6.71	7.17	6.17	7.58	6.17	6.17	6.17	6.17	7.21	6.17	6.17	6.17	13
	Overall rank	9	10	4.5	12	4.5	4.5	4.5	4.5	11	4.5	4.5	4.5	13
3	Boats	5.5	5.5	5.5	11	5.5	5.5	5.5	5.5	5.5	5.5	5.5	12	13
	Pepper	2.5	6	5	9	2.5	11	7	2.5	12	8	2.5	10	13
	Goldhill	5	5	5	11	5	5	11	5	5	5	11	5	13
	Lena	9	11	13	10	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	12
	House	11	5	11	11	5	5	5	5	5	5	5	5	13
	12,003	4.5	4.5	4.5	9	4.5	11	4.5	4.5	4.5	12	4.5	10	13
	181,079	5	12	5	11	5	5	5	5	5	5	5	10	13
	175,043	9.5	11	9.5	12	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	13
	101,085	5	11	10	12	5	5	5	5	5	5	5	5	13
	147,091	9	4.5	10.5	12	4.5	4.5	4.5	4.5	4.5	4.5	4.5	10.5	13
	101,087	5.5	12	5.5	11	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	13
	203,027	5	10.5	5	12	5	5	5	5	5	5	5	10.5	13
	Average rank	6.38	8.17	7.46	10.92	4.71	5.96	5.58	4.71	5.5	5.79	5.21	7.71	12.92
	Overall rank	8	11	9	12	1.5	7	5	1.5	4	6	3	10	13
4	Boats	6.5	10	9	11	8	3	3	3	3	6.5	3	12	13
	Pepper	12	4	10	11	8	4	4	4	4	9	4	4	13
	Goldhill	6.5	3	3	8.5	3	6.5	8.5	3	12	10	3	11	13
	Lena	9	4	10	12	4	4	4	4	4	8	4	11	13
	House	10	4	11	12	4	4	8.5	4	4	8.5	4	4	13
	12,003	6	3	7.5	7.5	3	9	3	3	10	11	3	13	12
	181,079	8	10	9	11	3.5	3.5	3.5	3.5	3.5	7	3.5	13	12
	175,043	8	10	9	11	4	4	4	4	4	4	4	12	13
	101,085	7.5	7.5	10	11	5	5	5	2	2	9	2	12	13
	147,091	9	2.5	11	12	6	8	6	2.5	2.5	10	6	2.5	13
	101,087	9	2.5	10	11	6.5	2.5	6.5	2.5	6.5	6.5	2.5	12	13
	203,027	11	7.5	9	12	7.5	3.5	3.5	3.5	3.5	3.5	3.5	10	13
	Average rank	8.54	5.67	9.04	10.83	5.21	4.75	4.96	3.25	4.92	7.75	3.54	9.71	12.83
	Overall rank	9	7	10	12	6	3	5	1	4	8	2	11	13

Table 14 (continued)

K	Images	WOA	GWO	COA	BBO	TLBO	GSA	ICA	CS	BA	FF	PSO	DE	GA
5	Boats	10	7	8	11	6	3	3	3	3	9	3	12	13
	Pepper	7	3	4	5	1.5	9	10	1.5	11	12	8	6	13
	Goldhill	10	5	6	8	1.5	7	9	1.5	11	12	3.5	3.5	13
	Lena	9	7.5	10	11	7.5	3	3	3	3	12	3	6	13
	House	5.5	10	8	9	5.5	2	5.5	2	2	12	5.5	11	13
	12,003	7	8	9	10	5.5	5.5	2.5	2.5	2.5	13	2.5	12	11
	181,079	7	8	9	10	6	2.5	2.5	2.5	2.5	11	5	12	13
	175,043	7	2.5	11	12	9	2.5	7	2.5	2.5	10	5	7	13
	101,085	8	7	9	10	6	4.5	4.5	3	1	12	2	11	13
	147,091	8	2.5	11	10	6	6	6	2.5	2.5	12	2.5	9	13
	101,087	8	5	10	12	8	6	8	3.5	1.5	11	1.5	3.5	13
	203,027	12	6.5	9	10	2.5	2.5	6.5	5	2.5	11	2.5	8	13
	Average rank	8.21	6	8.67	9.83	5.42	4.46	5.63	2.71	3.75	11.42	3.67	8.42	12.83
	Overall rank	8	7	10	11	5	4	6	1	3	12	2	9	13

Table 15 Friedman rank for benchmark images concerning cross entropy objective function

K	Chi-square value	P-value
2	101.56	2.7535E-16
3	82.25	1.5325E-12
4	93.91	8.5848E-12
5	97.54	1.6831E-12

Table 16 Wilcoxon signed rank test results concerning cross entropy objective function

Comparison	P-value
CS versus WOA	2.2676E-06
CS versus GWO	1.7597E-05
CS versus COA	2.5339E-06
CS versus BBO	5.1372E-08
CS versus TLBO	7.8516E-04
CS versus GSA	0.0012
CS versus ICA	6.1035E-05
CS versus BA	0.0244
CS versus FF	1.8019E-05
CS versus PSO	0.1016
CS versus DE	8.2081E-06
CS versus GA	1.6306E-09

conclude that the CS algorithm presented the best performance for image segmentation in this objective function.

Table 15 shows that given the Friedman Test, the P-value confirms that there is a significant difference between the evaluated algorithms. Furthermore, the results of Wilcoxon signed rank test between the CS and other algorithms is shown in Table 16. As can be seen in the table, the CS algorithm has experienced a significant improvement compared with all other algorithms, except PSO with $\alpha = 0.05$ level of significance and PSO algorithm with $\alpha = 0.2$ level of significance.

4 Conclusion

Image segmentation is the process of dividing an image into some meaningful regions. Image thresholding is one of the most important techniques for image segmentation. Image thresholding segments an image using existing information in the histogram of the image. The traditional multilevel thresholding techniques are so time-consuming, especially when the number of the threshold values is high. Population-based metaheuristic (P-metaheuristics) algorithms can be used as an alternative technique to solve this problem. P-metaheuristic algorithms are iterative algorithms that use a population of solutions to solve an optimization problem. In this paper, eight recently developed P-metaheuristic algorithms have been applied for image thresholding. These algorithms are whale optimization algorithm, grey wolf optimizer, cuckoo optimization algorithm, biogeography-based optimization, teaching-learning-based optimization, gravitational search algorithm, imperialist competitive algorithm, and cuckoo search. A Kapur's entropy -based measure has been used as the objective function in all used P-metaheuristic algorithms.

For conducting a more comprehensive comparison, these eight P-metaheuristic algorithms were compared with five other P-metaheuristic algorithms. The performance of the algorithms were evaluated in terms of objective function value, PSNR, FSIM, SSIM, stability analysis, and the search ability on the higher dimension problems. In addition, Friedman and Wilcoxon signed rank tests were provided as the non-parametric statistical analysis. Friedman rank showed that cuckoo search obtained the best rank in all threshold values expect K = 20. Friedman test confirmed that a significant difference exists in the behavior of the algorithms. In addition, the results of Wilcoxon signed rank test between CS and other algorithms revealed that there is a significant difference between them. To increase the reliability of the results, the efficiency of the P-metaheuristic algorithms were evaluated on another objective function (cross entropy). The results showed a relatively high correlation with the results of Kapur's entropy.

For future studies, these P-metaheuristic algorithms can be applied to complex and real-time images in various applications, such as satellite image processing and medical image processing. In addition, other objective functions can be evaluated using these most recently P-metaheuristic algorithms. This work also can be extended for processing of color images.

Acknowledgements Authors are grateful to University of Kashan for supporting this work under grant No. 572086.

References

1. Sanei SHR, Fertig RS (2015) Uncorrelated volume element for stochastic modeling of microstructures based on local fiber volume fraction variation. Compos Sci Technol 117:191–198
2. Smistad E, Falch TL, Bozorgi M, Elster AC, Lindseth F (2015) Medical image segmentation on GPUs—a comprehensive review. Med Image Anal 20(1):1–18
3. Mizushima A, Lu R (2013) An image segmentation method for apple sorting and grading using support vector machine and Otsu's method. Comput Electr Agric 94:29–37
4. Sanei SHR, Barsotti EJ, Leonhardt D, Fertig RS III (2016) Characterization, synthetic generation, and statistical equivalence of composite microstructures. J Compos Mater. doi:10.1177/0021998316662133
5. Gong M, Liang Y, Shi J, Ma W, Ma J (2013) Fuzzy c-means clustering with local information and kernel metric for image segmentation. IEEE Trans Image Process 22(2):573–584
6. Mohamed NA, Ahmed MN, Farag (1999) A modified fuzzy c-mean in medical image segmentation. In: IEEE international conference on acoustics, speech, and signal processing, IEEE, pp 3429–3432
7. Cai W, Chen S, Zhang D (2007) Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recognit 40(3):825–838
8. Zhang W, Li R, Deng H, Wang L, Lin W, Ji S, Shen D (2015) Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. NeuroImage 108:214–224
9. Vicente S, Kolmogorov V, Rother C (2008) Graph cut based image segmentation with connectivity priors. In: IEEE conference on computer vision and pattern recognition, IEEE, pp 1–8
10. Gao H, Xu W, Sun J, Tang Y (2010) Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. IEEE Trans Instrum Meas 59(4):934–946
11. Mousavi rad SJ, Akhlaghian Tab F, Mollazade K (2011) Classification of rice varieties using optimal color and texture features and BP neural networks. In: 7th Iranian conference on machine vision and image processing, IEEE, pp 1–5
12. Bhandari A, Kumar A, Singh G (2015) Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. Expert Syst Appl 42(22):8707–8730
13. White JM, Rohrer GD (1983) Image thresholding for optical character recognition and other applications requiring character image extraction. IBM J Res Dev 27(4):400–411
14. Kayal D, Banerjee S (2014) A new dynamic thresholding based technique for detection of hard exudates in digital retinal fundus image. In: International conference on signal processing and integrated networks, IEEE, pp 141–144
15. Kittler J, Illingworth J (1986) Minimum error thresholding. Pattern Recognit 19(1):41–47
16. Wang S, Chung F-l, Xiong F (2008) A novel image thresholding method based on Parzen window estimate. Pattern Recognit 41(1):117–129
17. Dirami A, Hammouche K, Diaf M, Siarry P (2013) Fast multilevel thresholding for image segmentation through a multiphase level set method. Signal Process 93 (1):139–153
18. Rajinikanth V, Aashiha J, Atchaya A (2014) Gray-level histogram based multilevel threshold selection with bat algorithm. Int J Comput Appl 93(16):1–8
19. 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(3):1573–1601
20. Pun T (1980) A new method for grey-level picture thresholding using the entropy of the histogram. Signal Process 2(3):223–237
21. Kapur JN, Sahoo PK, Wong AK (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285
22. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13(6):3066–3091
23. Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 41(7):3538–3560
24. Sezgin M (2004) Survey over image thresholding techniques and quantitative performance evaluation. J Electr Imaging 13(1):146–168
25. Yin P-Y (1999) A fast scheme for optimal thresholding using genetic algorithms. Signal Process 72(2):85–95
26. Tao W-B, Tian J-W, Liu J (2003) Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. Pattern Recognit Lett 24(16):3069–3078
27. Maitra M, Chatterjee A (2008) A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst Appl 34(2):1341–1350
28. Chander A, Chatterjee A, Siarry P (2011) A new social and momentum component adaptive PSO algorithm for image segmentation. Expert Syst Appl 38(5):4998–5004
29. Liu Y, Mu C, Kou W, Liu J (2015) Modified particle swarm optimization-based multilevel thresholding for image segmentation. Soft Comput 19(5):1311–1327
30. Mlakar U, Potočnik B, Brest J (2016) A Hybrid differential evolution for optimal multilevel image thresholding. Expert Syst Appl 65:221–232
31. Ayala HVH, dos Santos FM, Mariani VC, dos Santos Coelho L (2015) Image thresholding segmentation based on a

novel beta differential evolution approach. Expert Syst Appl 42(4):2136–2142
32. Sarkar S, Das S (2013) Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach. IEEE Trans Image Process 22(12):4788–4797
33. Sathya P, Kayalvizhi R (2011) Modified bacterial foraging algorithm based multilevel thresholding for image segmentation. Eng Appl Artif Intell 24(4):595–615
34. Alihodzic A, Tuba M (2014) Improved bat algorithm applied to multilevel image thresholding. Sci World J 2014:1–16
35. Horng M-H (2011) Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst Appl 38(11):13785–13791
36. Sağ T, Çunkaş M (2015) Color image segmentation based on multiobjective artificial bee colony optimization. Appl Soft Comput 34:389–401
37. Cuevas E, Sención F, Zaldivar D, Pérez-Cisneros M, Sossa H (2012) A multi-threshold segmentation approach based on artificial bee colony optimization. Appl Intell 37(3):321–336
38. Horng M-H (2010) A multilevel image thresholding using the honey bee mating optimization. Appl Math Comput 215(9):3302–3310
39. Oliva D, Cuevas E, Pajares G, Zaldivar D, Osuna V (2014) A multilevel thresholding algorithm using electromagnetism optimization. Neurocomputing 139:357–381
40. Raja N, Rajinikanth V, Latha K (2014) Otsu based optimal multilevel image thresholding using firefly algorithm. Model Simul Eng
41. Fan C, Ouyang H, Zhang Y, Xiao L (2014) Optimal multilevel thresholding using molecular kinetic theory optimization algorithm. Appl Math Comput 239:391–408
42. Hammouche K, Diaf M, Siarry P (2010) A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. Eng Appl Artif Intell 23(5):676–688
43. Cuevas E, Sossa H (2013) A comparison of nature inspired algorithms for multi-threshold image segmentation. Expert Syst Appl 40(4):1213–1219
44. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
45. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
46. Rajabioun R (2011) Cuckoo optimization algorithm. Appl Soft Comput 11(8):5508–5518
47. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
48. Rao RV, Savsani VJ, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
49. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf sci 179(13):2232–2248
50. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on evolutionary computation, IEEE, pp 4661–4667
51. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: World congress on nature and biologically inspired computing, IEEE, pp 210–214
52. Precup R-E, David R-C, Petriu EM, Szedlak-Stinean A-I, Bojan-Dragos C-A (2016) Grey wolf optimizer-based approach to the tuning of pi-fuzzy controllers with a reduced process parametric sensitivity. IFAC Papers Online 49(5):55–60
53. Mirjalili S, Saremi S, Mirjalili SM, Coelho LdS (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl 47:106–119
54. Medjahed S, Saadi TA, Benyettou A, Ouali M (2016) Gray wolf optimizer for hyperspectral band selection. Appl Soft Comput 40:178–186
55. Sharma R, Rana K, Kumar V (2014) Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator. Expert Syst Appl 41(9):4274–4289
56. Azizipanah-Abarghooee R, Niknam T, Zare M, Gharibzadeh M (2014) Multi-objective short-term scheduling of thermoelectric power systems using a novel multiobjective θ-improved cuckoo optimisation algorithm. IET Gener Transm Distrib 8(5):873–894
57. Mehdinejad M, Mohammadi-Ivatloo B, Dadashzadeh-Bonab R (2016) Energy production cost minimization in a combined heat and power generation systems using cuckoo optimization algorithm. Energy Effic 10:1–16
58. Farshchin M, Camp C, Maniat M (2016) Multi-class teaching–learning-based optimization for truss design with frequency constraints. Eng Struct 106:355–369
59. Sahu RK, Panda S, Rout UK, Sahoo DK (2016) Teaching learning based optimization algorithm for automatic generation control of power system using 2-DOF PID controller. Int J Electr Power Energy Syst 77:287–301
60. Shuaib YM, Kalavathi MS, Rajan CCA (2015) Optimal capacitor placement in radial distribution system using gravitational search algorithm. Int J Electr Power Energy Syst 64:384–397
61. Gouthamkumar N, Sharma V, Naresh R (2015) Disruption based gravitational search algorithm for short term hydrothermal scheduling. Expert Syst Appl 42(20):7000–7011
62. Beigvand SD, Abdi H, La Scala M (2016) Optimal operation of multicarrier energy systems using time varying acceleration coefficient gravitational search algorithm. Energy 114:253–265
63. Fowlkes C, Martin D, Malik J (2012) The berkeley segmentation dataset and benchmark (bsdb). http://www.cs.berkeley.edu/projects/vision/grouping/segbench. Accessed 11 June 2017
64. Simon D (2008) Biogeography-based optimization. Evol Comput IEEE Trans 12(6):702–713
65. Hosseini S, Al Khaled A (2014) A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research. Appl Soft Comput 24:1078–1094
66. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
67. Yang X-S (2010) Firefly algorithm, Levy flights and global optimization. In: Research and development in intelligent systems XXVI. Springer, pp 209–218
68. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Report 2005005:2005
69. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
70. Zhang L, Zhang L, Mou X, Zhang D (2011) FSIM: a feature similarity index for image quality assessment. IEEE Trans Image Process 20(8):2378–2386
71. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
72. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18