Egyptian Informatics Journal (2015) 16, 133–150

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Fuzzy 2-partition entropy threshold selection based on Big Bang–Big Crunch Optimization algorithm

Baljit Singh Khehra [a,*], Amar Partap Singh Pharwaha [b,1], Manisha Kaushal [c,2]

[a] *Computer Science & Engineering Department, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib 140407, Punjab, India*
[b] *Electronics & Communication Department, Sant Longowal Institute of Engineering & Technology, Deemed University (Established by Govt. of India), Longowal-148106, Sangrur, Punjab, India*
[c] *Computer Science & Engineering, UIET, Punjab University, Chandigarh 160014, India*

Received 5 September 2014; revised 3 February 2015; accepted 16 February 2015
Available online 10 March 2015

KEYWORDS

Big Bang–Big Crunch Optimization;
Biogeography-based Optimization;
Fuzzy 2-partition entropy;
Optimal threshold;
Image segmenting

Abstract The fuzzy 2-partition entropy approach has been widely used to select threshold value for image segmenting. This approach used two parameterized fuzzy membership functions to form a fuzzy 2-partition of the image. The optimal threshold is selected by searching an optimal combination of parameters of the membership functions such that the entropy of fuzzy 2-partition is maximized. In this paper, a new fuzzy 2-partition entropy thresholding approach based on the technology of the Big Bang–Big Crunch Optimization (BBBCO) is proposed. The new proposed thresholding approach is called the BBBCO-based fuzzy 2-partition entropy thresholding algorithm. BBBCO is used to search an optimal combination of parameters of the membership functions for maximizing the entropy of fuzzy 2-partition. BBBCO is inspired by the theory of the evolution of the universe; namely the Big Bang and Big Crunch Theory. The proposed algorithm is tested on a number of standard test images. For comparison, three different algorithms included Genetic Algorithm (GA)-based, Biogeography-based Optimization (BBO)-based and recursive approaches are also implemented. From experimental results, it is observed that the performance of the proposed algorithm is more effective than GA-based, BBO-based and recursion-based approaches.

* Corresponding author. Tel.: +91 9463446505/+91 8283809605.
E-mail addresses: baljitkhehra74@gmail.com, baljitkhehra@rediffmail.com, baljit.singh@bbsbec.ac.in (B.S. Khehra), amarpartapsingh@yahoo.com (A.P.S. Pharwaha), er.manishakaushal@gmail.com (M. Kaushal).
1 Tel.: +91 9463122255.
2 Tel.: +91 9592200491.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

1. Introduction

Image segmentation is a fundamental pre-processing technique in the field of machine vision. The purpose of image segmentation is to subdivide an image into non-overlapping constituent regions that have the same features and to extract those of interest. Depending on the particular application, different techniques for image segmentation have been used, such as image thresholding, edge detection, region growing, stochastic models, ANN and clustering techniques [1]. Image thresholding is one of the well-known techniques for image segmentation

http://dx.doi.org/10.1016/j.eij.2015.02.004
1110-8665 © 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

because of its simplicity and stable performance among other image segmentation techniques. One of the most frequently used image thresholding approaches is global thresholding [2]. The principal assumption behind global thresholding is that the foreground and background regions can be distinguished by selecting a suitable value from gray level values of image known as threshold value. In the past years, many global thresholding techniques have been proposed by researchers to select suitable threshold value for segmenting images and recognize patterns. Pun [3] proposed a method for image thresholding. This method is based on Shannon entropy. Kapur [4] found a method for gray-level picture thresholding using the entropy of the histogram. In this method, Kapur's measure is used to find the entropy of image histogram. Sahoo et al. [5] used Renyi's entropy to find the threshold value of the image. De Albuquerque et al. [6] proposed a thresholding method based on Tsallis entropy. Li and Lee [7] solved the threshold selection problem by minimizing the crossentropy between the original image and its segmented version. Nie et al. [8] used co-occurrence matrix to find threshold value in 2D minimum local crossentropy thresholding approach. Tang et al. [9] employed genetic algorithm to select the thresholds in multilevel minimum crossentropy threshold selection approach. Horng [10] used the honeybee mating optimization to find the thresholds of the multilevel thresholding. Horng and Liou [11] applied the firefly algorithm to search for the thresholds of histogram of image based on the minimum crossentropy criterion.

Cheng et al. [12] proposed fuzzy c-partition entropy approach to select threshold value. In this method, two parameterized fuzzy membership functions are used to form a fuzzy 2-partition of the image. One partition is related to the foreground region and other is related to the background region. The optimal threshold is selected by searching an optimal combination of parameters of the membership functions such that the entropy of fuzzy 2-partition is maximized. Benabdelkader and Boulemden [13] used a recursive algorithm to find a suitable combination of parameters of the membership functions. In this approach, two parameters of trapezium fuzzy membership function are optimized to find suitable threshold value. In Tang et al. [14] recursive approach, three parameters of two fuzzy membership functions (s-function and z-function) are optimized to select optimal threshold value of image. Despite a large number of researchers have done work in the field of image thresholding by using fuzzy c-partition entropy, they all face the problem that the computation time needed for searching an optimal combination of parameters of the membership functions to maximize the entropy of fuzzy c-partition increases exponentially when the number of parameters of the membership functions increases. Thus, selection of an optimal combination of parameters of the membership functions to maximize the entropy of fuzzy c-partition is an optimization problem. In last decades, metaheuristic algorithms have been flourishing for optimization problems. These algorithms are inspired by the behaviors of natural phenomena. Many different metaheuristic algorithms such as genetic algorithm [15], ant colony optimization algorithm [16], biogeography based optimization approach [17], bacterial foraging algorithm [18] have been successfully applied to maximize entropy of fuzzy c-partition for image segmenting. Big Bang–Big Crunch Optimization (BBBCO) algorithm is a recent member in the optimal algorithm family [19]. It is inspired by the theory of the evolution of the universe, namely the Big Bang and Big Crunch Theory. The search mechanism of BBBCO generates random points in the Big Bang phase and shrinks those points to a single representative point via a center of mass or maximum cost approach in the Big Crunch phase. BBBCO algorithm has several advantages over other metaheuristic algorithms: the inherent numerical simplicity of the algorithm with relatively few control parameters, low computation cost, high convergence, and easy implementation [20]. For its simplicity, easy implementation, low time complexity, quick convergence, quality of solution, and robustness, recently BBBCO algorithm has been successfully applied to a variety of optimization problems, such as parameter estimation in structural systems [20], design of steel frames [21], optimized echo state networks [22], and fuzzy controllers design [23,24]. Encouraged by these successful applications, the feasibility of BBBCO algorithm for searching an optimal combination of parameters of the fuzzy membership functions to find optimal threshold value in the image is investigated.

The aim of this paper was focused on the selection of optimal image threshold value by using fuzzy 2-partition entropy criterion. In this paper, BBBCO algorithm is used to select an optimal combination of parameters of the membership functions for maximizing the entropy of fuzzy 2-partition. The selected optimal parameters are used to find optimal image threshold value. The proposed thresholding approach is called the BBBCO-based fuzzy 2-partition entropy thresholding algorithm. The performance of proposed algorithm is tested on a number of standard test images. For comparison, three different algorithms included Genetic Algorithm (GA)-based, Biogeography-based Optimization (BBO)-based and recursion-based approaches are also implemented.

The rest of the paper is organized as follows. Section 2 introduces the Big Bang–Big Crunch Optimization algorithm. Section 3 presents the mathematical model used to describe the concepts of fuzzy 2-partition entropy for image. In Section 4, the proposed BBBCO-based fuzzy 2-partition entropy thresholding algorithm is described in detail. Experiments on 11 standard test images are carried out in Section 5, where comparisons with GA-based, BBO-based, recursion-based approaches and the proposed approach are made. Finally, Section 6 concludes the paper and suggests future research directions.

2. Big Bang–Big Crunch Optimization algorithm

The Big Bang–Big Crunch theory is a broadly accepted theory for the origin and evolution of the universe. In the Big Bang phase, energy dissipation produces disorder state of particles randomly, whereas, in the Big Crunch phase, randomly distributed particles are drawn into an order. The Big Bang–Big Crunch Optimization (BBBCO) algorithm is inspired by this theory [19]. Randomness is the key feature of the Big Bang phase. Randomness resembles the energy dissipation in nature while convergence to a local or global optimum point can be seen as gravitational attraction [20]. Since energy dissipation creates disorder from ordered particles, randomness is used as a transformation from a converged solution (order) to the birth of totally new candidate solutions (disorder).

BBBCO algorithm consists of two phases: Big Bang phase and Big Crunch phase. Initially, a population of some candidate solutions is created from the entire search space. In Big

Bang phase, candidate solutions are randomly distributed over the entire search space. The Big Bang phase is followed by the Big Crunch phase. The Big Crunch is a convergence operator that has many inputs but only one output, which can be named as the center of mass. The center of mass is computed from the current positions of each candidate solution in the population and its associated fitness value. It is defined as

$$C = \frac{\sum_{i=1}^{N_p} \frac{x_i}{F_i}}{\sum_{i=1}^{N_p} \frac{1}{F_i}} \tag{1}$$

where C is the position of the center of mass; x_i is the position of candidate i in an n-dimensional search space; F_i is a fitness value of candidate i; and N_p is the population size in Big Bang phase. Here, n is the number of parameters to be optimized.

The new candidates around the center of mass are calculated as follows:

$$x^{new} = C + r(x_{\max} - x_{\min})/(k+1) \tag{2}$$

where r is a random number in the interval [−1, 1]; $x_{\max}$ and $x_{\min}$ are the upper and lower limits on the values of the candidates; and k is the current iteration step.

Based upon the above discussion, BBBCO algorithm can be re-written in the following steps [23]:

Step 1: [Initialization] Generate a random set of N_p candidate solutions (population) in the search space.
Step 2: [Fitness Evaluation] Evaluate the fitness function values of all the candidate solutions of the population.
Step 3: [Big Crunch Phase] Calculate the center of mass for the current population of candidate solutions using Eq. (1).
Step 4: [Big Bang Phase] Calculate new candidate solutions around the center of mass using Eq. (2).
Step 5: Return to Step 2 until the stopping criteria (maximum number of iterations/desired solution) is not met.
Step 6: Stop.

3. Fuzzy 2-partition entropy of image

Let $f(x, y)$ be the gray level value of the image at the pixel (x, y) and $p_0, p_1, p_2, \ldots, p_j, \ldots, p_{L_{\max}}$ be the probability distribution of the gray level values of the image. p_j is defined as

$$p_j = n_j/(M \times N) \tag{3}$$

where n_j is the number of pixels in the image with gray level j, $j = 0, 1, 2, \ldots, L_{\max}$.

For partitioning an image into foreground (white) and background (black) regions, two fuzzy membership functions (S-function and Z-function) [25] are used. S-fuzzy membership function is a membership function of the foreground region and Z-fuzzy membership function is a membership function of the background region. Such partitioning is called fuzzy 2-partition of image. S-fuzzy membership function is defined as

$$\mu_F(j) = S(j; a, b, c) = \begin{cases} 0 & j \leqslant a \\ \frac{(j-a)^2}{(c-a)(b-a)} & a < j \leqslant b \\ 1 - \frac{(j-c)^2}{(c-a)(c-b)} & b < j \leqslant c \\ 1 & j > c \end{cases} \tag{4}$$

and Z-fuzzy membership function is defined as

$$\mu_B(j) = Z(j; a, b, c) = \begin{cases} 1 & j \leqslant a \\ 1 - \frac{(j-a)^2}{(c-a)(b-a)} & a < j \leqslant b \\ \frac{(j-c)^2}{(c-a)(c-b)} & b < j \leqslant c \\ 0 & j > c \end{cases} \tag{5}$$

where $0 \leqslant a \leqslant b \leqslant c \leqslant L_{\max}$, parameters a, b and c determine the shape of S and Z fuzzy membership functions as shown in Fig. 1.

Two probability distribution functions can be found from the above mentioned fuzzy distribution of gray level values of the image: one for the foreground region and second for the background region. The probability distribution functions of the foreground and the background regions [26] are described as follows:

Probability distribution function of the foreground region:

$$P_F = \sum_{j=0}^{L_{\max}} \mu_F(j) p(j) \tag{6}$$

Probability distribution function of the background region:

$$P_B = \sum_{j=0}^{L_{\max}} \mu_B(j) p(j) \tag{7}$$

From the definition of Shannon entropy [27], the entropy of foreground region pixels and the background region pixels can be defined as follows:

Shannon fuzzy entropy for the foreground region pixels:

$$H_F(a, b, c) = -\sum_{j=0}^{L_{\max}} \mu_F(j) p_j \log\left(\sum_{j=0}^{L_{\max}} \mu_F(j) p_j\right) \tag{8}$$

Shannon fuzzy entropy for the background region pixels:

$$H_B(a, b, c) = -\sum_{j=0}^{L_{\max}} \mu_B(j) p_j \log\left(\sum_{j=0}^{L_{\max}} \mu_B(j) p_j\right) \tag{9}$$

The sum of the two entropies is written as

$$H(a, b, c) = H_F(a, b, c) + H_B(a, b, c) \tag{10}$$

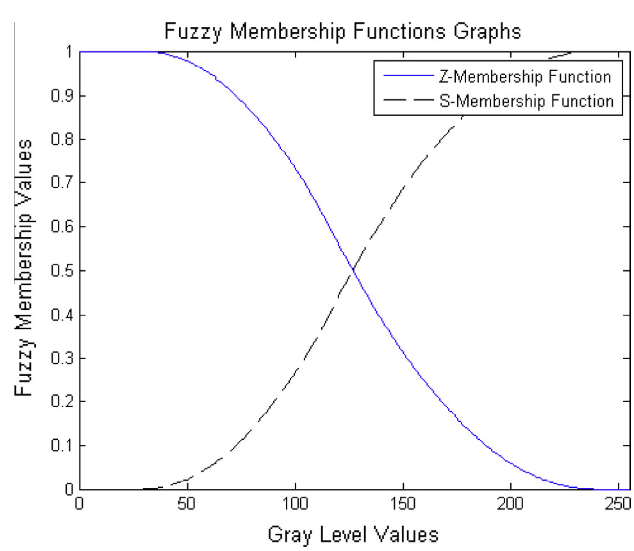

Figure 1 Graphical representation of Z and S fuzzy membership functions.

The objective is to search an optimal combination of parameters a, b and c such that H (a, b, c) is maximized. The optimal values of a, b and c that maximize H (a, b, c) are used to find optimal threshold for separating the foreground region from the image background [14] i.e.

$$(a^*, b^*, c^*) = \underset{(a,b,c)\in f}{\arg\max}\{H(a,b,c)\} \tag{11}$$

where f is the gray level value of the image whose value varies from 0 to $L_{\max}$.

Threshold value is the intersection point of μ_F and μ_B curves as shown in Fig. 1. The intersection point depends upon the parameters a, b and c. It is calculated from Eqs. (4) and (5). Thus, the optimal threshold from the optimal values of a, b and c that maximize H (a, b, c) is calculated as follows [14,16]:

$$T = \begin{cases} a^* + \sqrt{(c^* - a^*)(b^* - a^*)/2} & (a^* + c^*)/2 \leqslant b^* \leqslant c^* \\ c^* - \sqrt{(c^* - a^*)(c^* - b^*)/2} & a^* \leqslant b^* \leqslant (a^* + c^*)/2 \end{cases} \tag{12}$$

4. BBBCO-based fuzzy 2-partition entropy thresholding algorithm

The parameter estimation problem is an optimization problem in which the goal is to search an optimal combination of parameters (a, b, c) such that Eq. (11) is maximized. Thus, optimization problem is defined as follows:

Find an optimal combination of a, b, c such that H (a, b, c) is maximum. Here, objective function is

$$H(a,b,c) = -\left[\sum_{j=0}^{L_{\max}} \mu_F(j)p_j \log\left(\sum_{j=0}^{L_{\max}} \mu_F(j)p_j\right)\right] - \left[\sum_{j=0}^{L_{\max}} \mu_B(j)p_j \log\left(\sum_{j=0}^{L_{\max}} \mu_B(j)p_j\right)\right] \tag{13}$$

In case of gray level images that have 256 gray levels, the search space of parameters is as follows:

$(a,b,c) \in [0, 255]$

In this paper, a BBBCO-based fuzzy 2-partition entropy thresholding algorithm is proposed. The detail of the proposed algorithm is introduced as follows:

Step 1: Initialize the optimization parameters

- Population size (N_p)
- Number of iterations (I_n)
- Number of variables (var = 3: a, b, c)
- Limits of parameters ($0 \leqslant a, b, c \leqslant L_{\max}$)

Step 2: Initialize the population

Generate random population according to the population size and the number of parameters.

$Pop = X[N_p, \text{var}]$

Population is a random set of some feasible solution candidates. Each feasible solution candidate is denoted as

$S_i \leftarrow X[i, \text{var}]$

$S_i^k = [a_i \quad b_i \quad c_i],\ i = 1, 2, 3, \ldots, N_p$

Step 3: Evaluate the value of the objective function for each feasible solution candidate as its fitness

$F_i,\ i = 1, 2, 3, \ldots, N_p$

The following procedure is adopted for this evaluation:

Step 3.1: Read input image I

Step 3.2: Find the size of input image $M \times N$

Step 3.3: Find the number of pixels in the image whose gray level j, j = 0, 1, 2, ..., $L_{\max}$

$h(j),\ j = 0, 1, 2, \ldots, L_{\max}$

Step 3.4: Find the probability distribution of the gray level values of the image

$$p(j) = h(j)/(M \times N) \tag{14}$$

Step 3.5: Fuzzification of input image I through parameters a_i, b_i and c_i using Eqs. (4) and (5)

For each gray level value j, j = 0, 1, 2, ..., $L_{\max}$

$$\mu_F(j) = \begin{cases} 0 & j \leqslant a_i \\ \frac{(j-a_i)^2}{(c_i-a_i)(b_i-a_i)} & a_i < j \leqslant b_i \\ 1 - \frac{(j-c_i)^2}{(c_i-a_i)(c_i-b_i)} & b_i < j \leqslant c_i \\ 1 & j > c_i \end{cases} \tag{15}$$

$$\mu_B(j) = \begin{cases} 1 & j \leqslant a_i \\ 1 - \frac{(j-a_i)^2}{(c_i-a_i)(b_i-a_i)} & a_i < j \leqslant b_i \\ \frac{(j-c_i)^2}{(c_i-a_i)(c_i-b_i)} & b_i < j \leqslant c_i \\ 0 & j > c_i \end{cases} \tag{16}$$

Step 3.6: Find fuzzy entropy for the foreground region and the background region using Eqs. (8) and (9)

$$H_F(a_i, b_i, c_i) = -\sum_{j=0}^{L_{\max}} \mu_F(j)p(j) \log\left(\sum_{j=0}^{L_{\max}} \mu_F(j)p(j)\right) \tag{17}$$

$$H_B(a_i, b_i, c_i) = -\sum_{j=0}^{L_{\max}} \mu_B(j)p(j) \log\left(\sum_{j=0}^{L_{\max}} \mu_B(j)p(j)\right) \tag{18}$$

Step 3.7: Evaluate the fitness value of a feasible solution candidate of population in an arbitrary iteration using Eq. (10)

$$F_i = H(S_i) = H(a_i, b_i, c) = H_F(a_i, b_i, c_i) + H_B(a_i, b_i, c_i) \tag{19}$$

Step 4: Arrange feasible solution candidates of the population in descending order according to their respective fitness value

S_i and the corresponding fitness function value F_i

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ . & . & . \\ . & . & . \\ X_{N_p1} & X_{N_p2} & X_{N_p3} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ . \\ . \\ S_{N_p} \end{bmatrix}$$ and the corresponding fitness value

$$F = \begin{bmatrix} F_1 \\ F_2 \\ . \\ . \\ F_{N_p} \end{bmatrix}$$

$\{X_{i1}\} \in a, \{X_{i2}\} \in b, \{X_{i3}\} \in c$

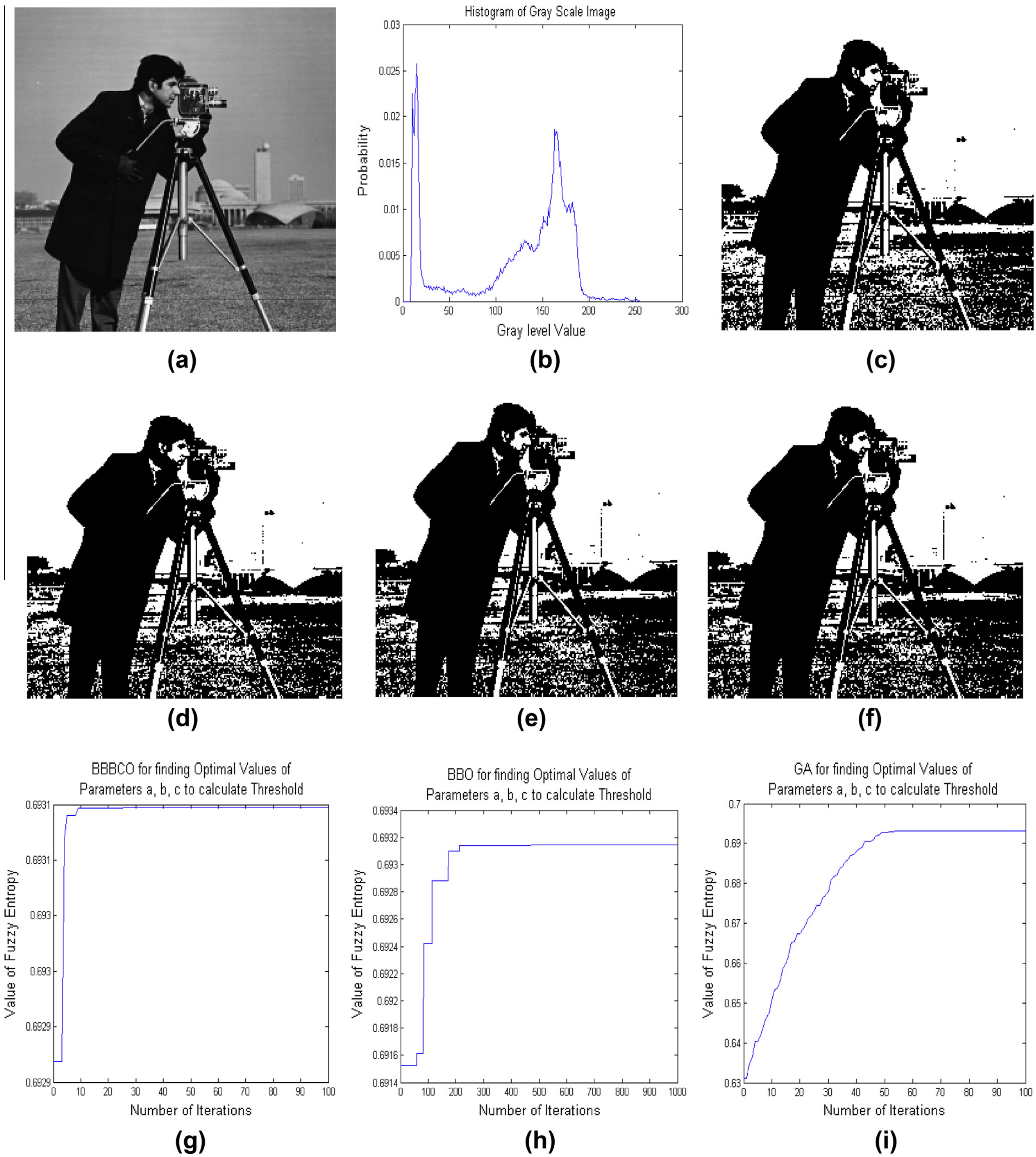

Figure 2 Segmentation of cameraman image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

Step 5: For guiding the new search space, the center of mass for the current population of the feasible solution candidates at the k^{th} iteration is determined as

$$C_j^k = \frac{\sum_{i=1}^{N_p} \frac{X_{ij}^k}{F_i^k}}{\sum_{i=1}^{N_p} \frac{1}{F_i^k}} \tag{20}$$

X_{ij}^k is the value of the j^{th} variable (j = 1, 2, 3) of the ith feasible solution candidate (i = 1, 2, 3, ..., N_p) at the kth iteration (k = 1, 2, 3, ..., I_n).

Step 6: Generate $N_p - 1$ feasible solution candidates for the next population around the center of mass by adding or subtracting a normal random number whose value decreases as the iteration elapses. This can be formulated as

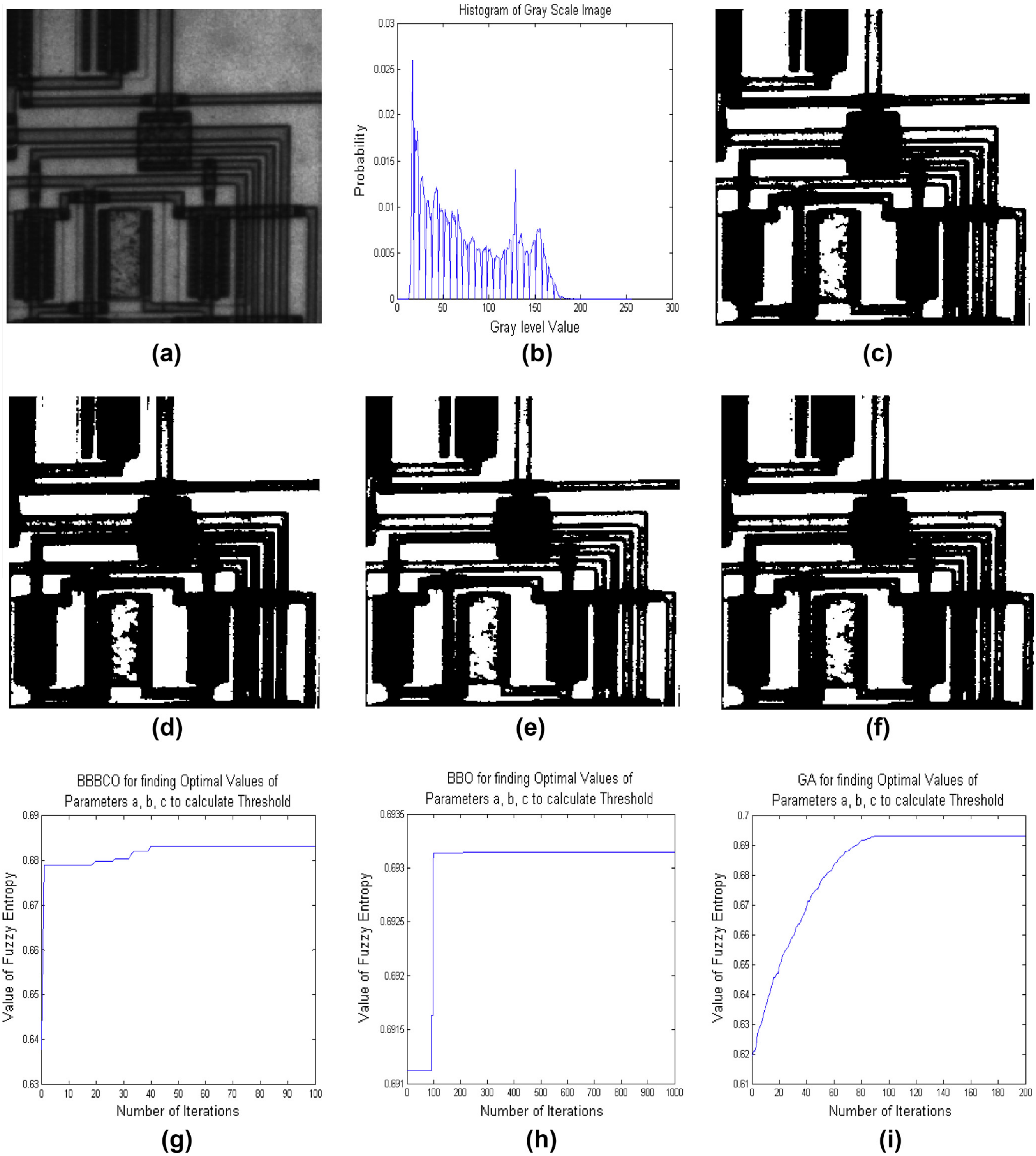

Figure 3 Segmentation of circuit image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

$$X_{ij}^{k+1} = C_j^k + r_j\left(X_{ij}^{\max} - X_{ij}^{\min}\right)/(k+1) \tag{21}$$

where C_j^k stand for the center of mass for the variable j at kth iteration, r_j is a random number from [−1, 1] for the variable j, $X_{ij}^{\max}$ is the upper limit of the variable j and $X_{ij}^{\min}$ is the lower limit of variable j.

Step 7: Generate new population (X^{k+1}) by replacing $N_p - 1$ feasible solution candidates of the old population (X^k) with $N_p - 1$ feasible solution candidates generated in Step 6. The best one feasible solution candidate of the old

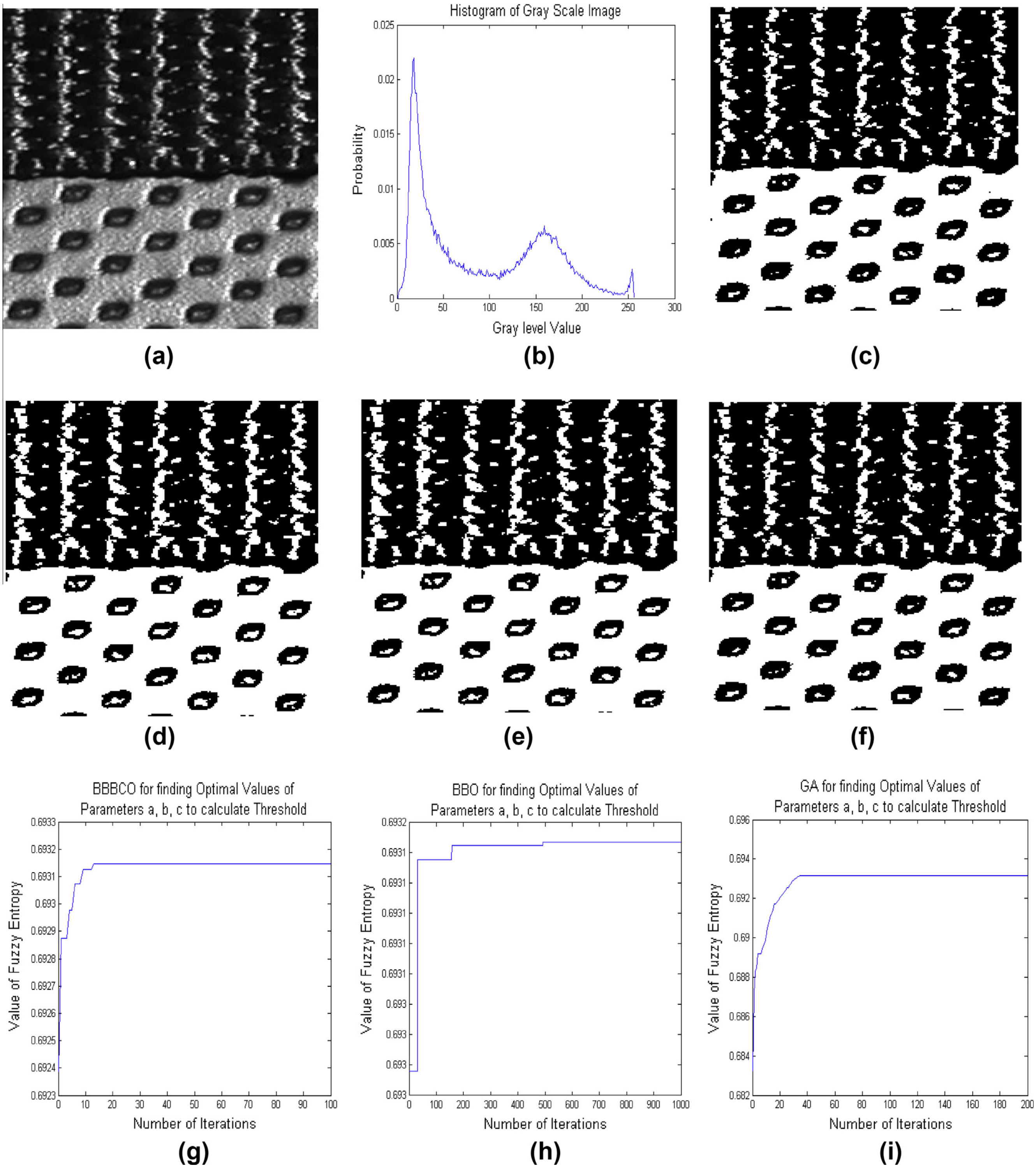

Figure 4 Segmentation of bag image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

population (X^k) is copied to new population (X^{k+1}). This is done to prevent loosing the best-found solution.

Step 8: Return to Step 3 until the stopping criteria (maximum iteration/desired solution) is not met.

Step 9: Stop.

5. Experimental results and comparative performance

In this section, the experimental results are discussed. This discussion includes the selection of optimal threshold value, and some experiments are performed on standard test images to

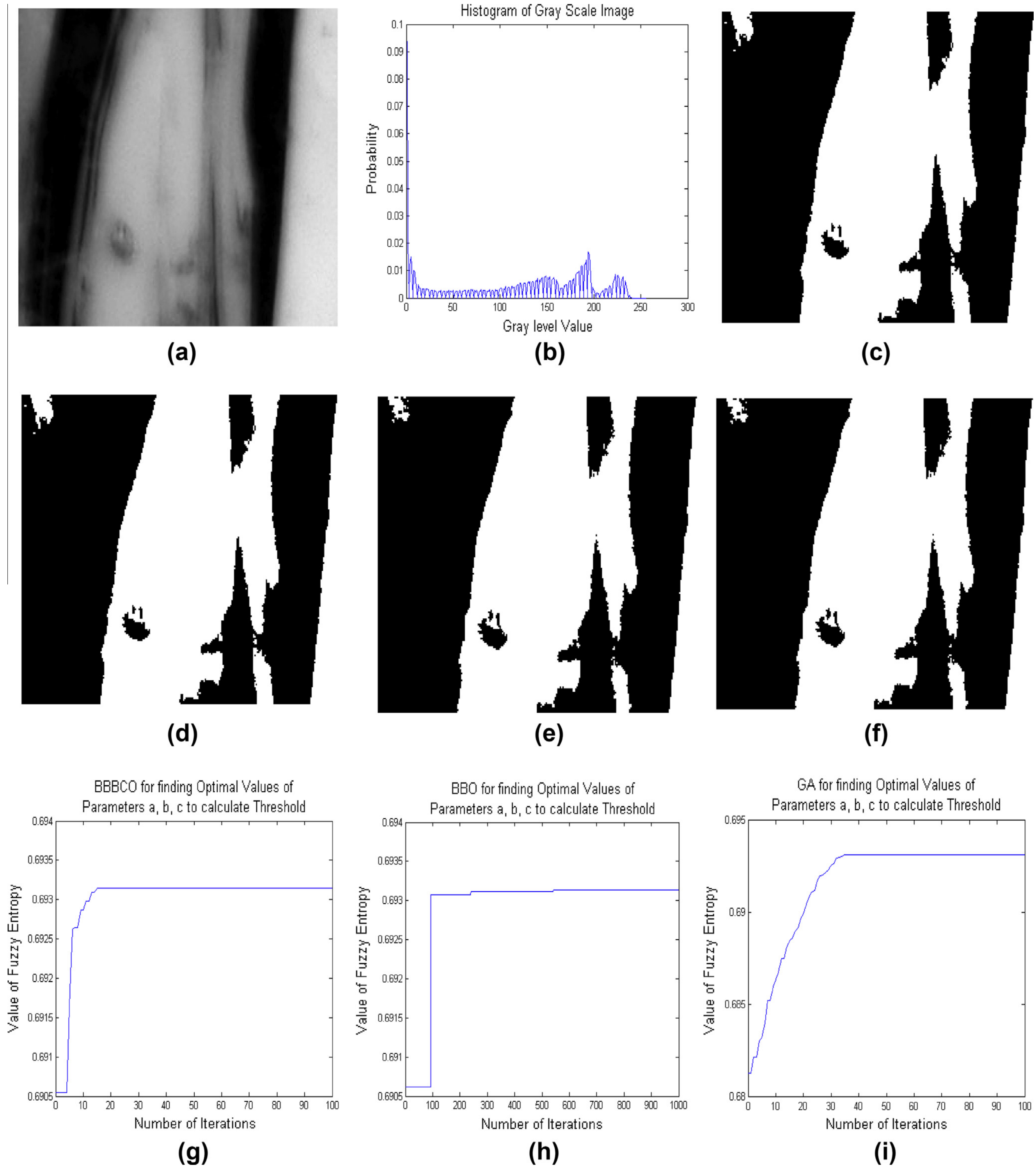

Figure 5 Segmentation of glass image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (*a*, *b*, *c*) that are used for calculating optimal threshold.

illustrate the performance of the proposed approach for thresholding. For comparison, BBO-based fuzzy 2-partition entropy thresholding algorithm, GA-based fuzzy 2-partition entropy thresholding algorithm and recursion-based fuzzy 2-partition entropy thresholding algorithm are considered. MATLAB 7.7.0 (R2008b) is used to implement these algorithms. All the experiments are performed on Intel(R) Core(TM) 2 Duo CPU T7500, 1.99 GB of RAM with

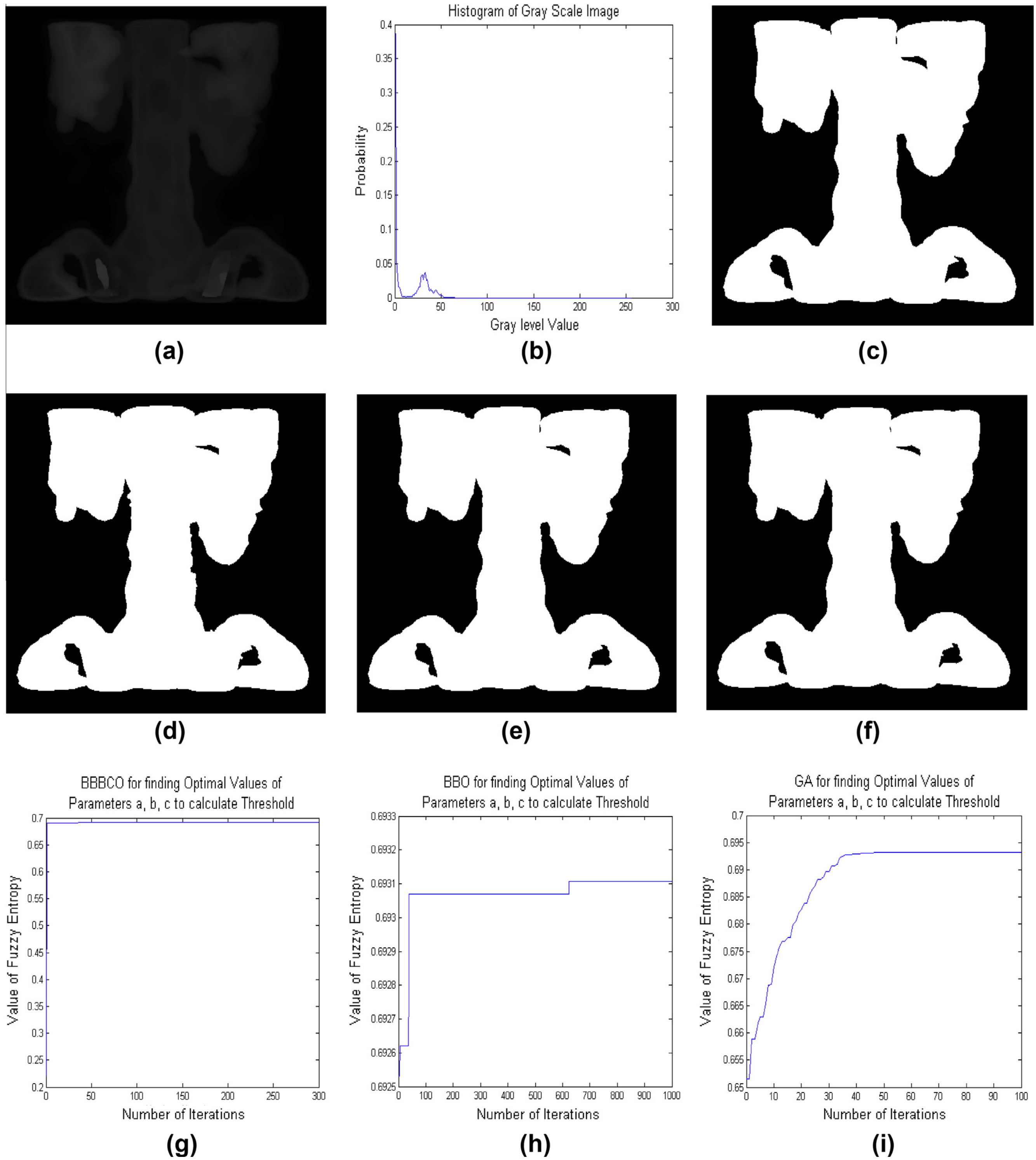

Figure 6 Segmentation of spine image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

2.2 GHz machine. The proposed algorithm, BBO-based fuzzy 2-partition entropy thresholding algorithm and GA-based fuzzy 2-partition entropy thresholding algorithm are meta-heuristic algorithms while recursion-based fuzzy 2-partition entropy thresholding algorithm is a recursive algorithm. The key parameters of meta-heuristic algorithms are listed below:

BBBCO-based algorithm: population size = 10, elitism parameter = 1, maximum number of iteration = 100.
BBO-based algorithm: population size = 10, mutation rate = 0.15, maximum immigration rate = 1, maximum emigration rate = 1, elitism parameter = 1, maximum number of iteration = 1000.

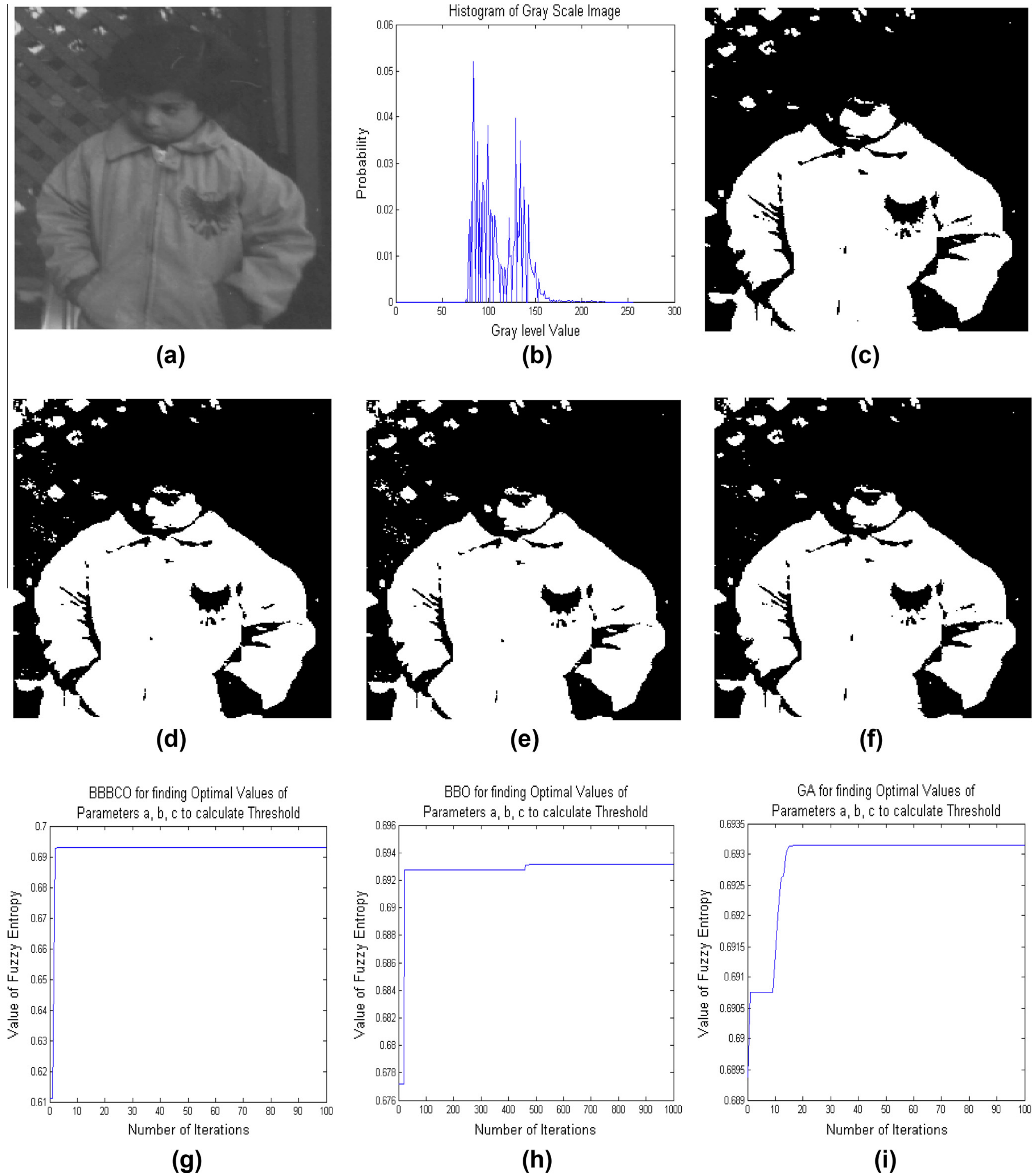

Figure 7 Segmentation of pout image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

GA-based algorithm: population size = 10, crossover probability = 0.75, mutation probability = 0.15, elitism parameter = 1, maximum number of iteration = 100.

5.1. Experimental results

Eleven standard test images, named “Cameraman”, “Circuit”, “Bag”, “Glass”, “Spine”, “Pout”, “Moon_surface”, “Lion”,

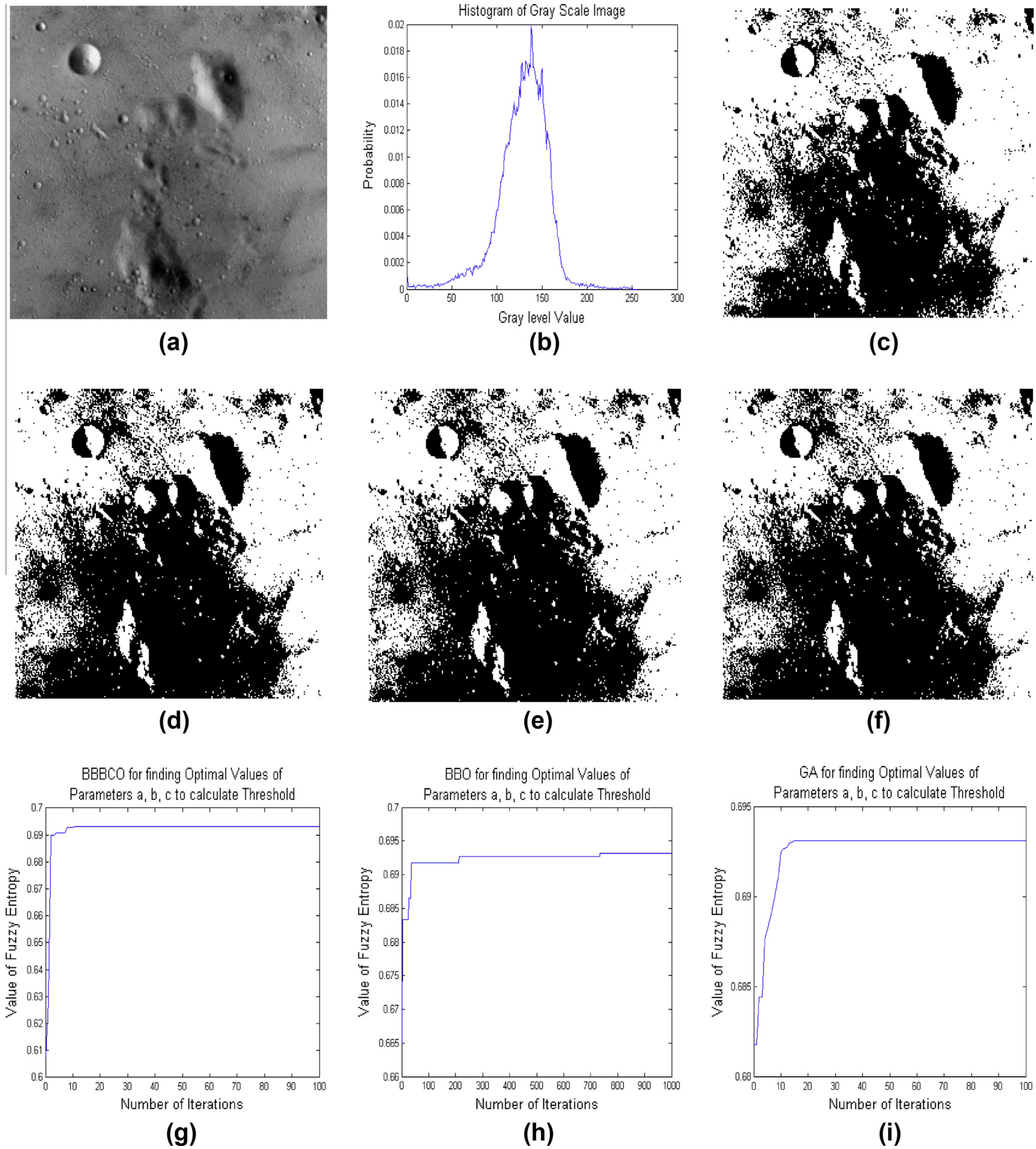

Figure 8 Segmentation of moon_surface image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

"Butterfly", "US_air_force_plane" and "Lady" with size 256×256, 280×272, 250×189, 181×282, 367×490, 291×240, 256×256, 512×512, 512×512, 512×512 and 512×512, respectively, and with 256 gray levels and different histogram distribution. The standard test images are shown in Figs. 2–12(a) and their histograms are shown in Figs. 2–12(b). The optimal values of parameters a, b and c are computed by four methods for these eleven images. Table 1 shows the optimal values of parameters a, b and c that are found for eleven standard test images by different methods. The optimal threshold values are calculated from the optimal values of parameters a, b and c obtained by different methods

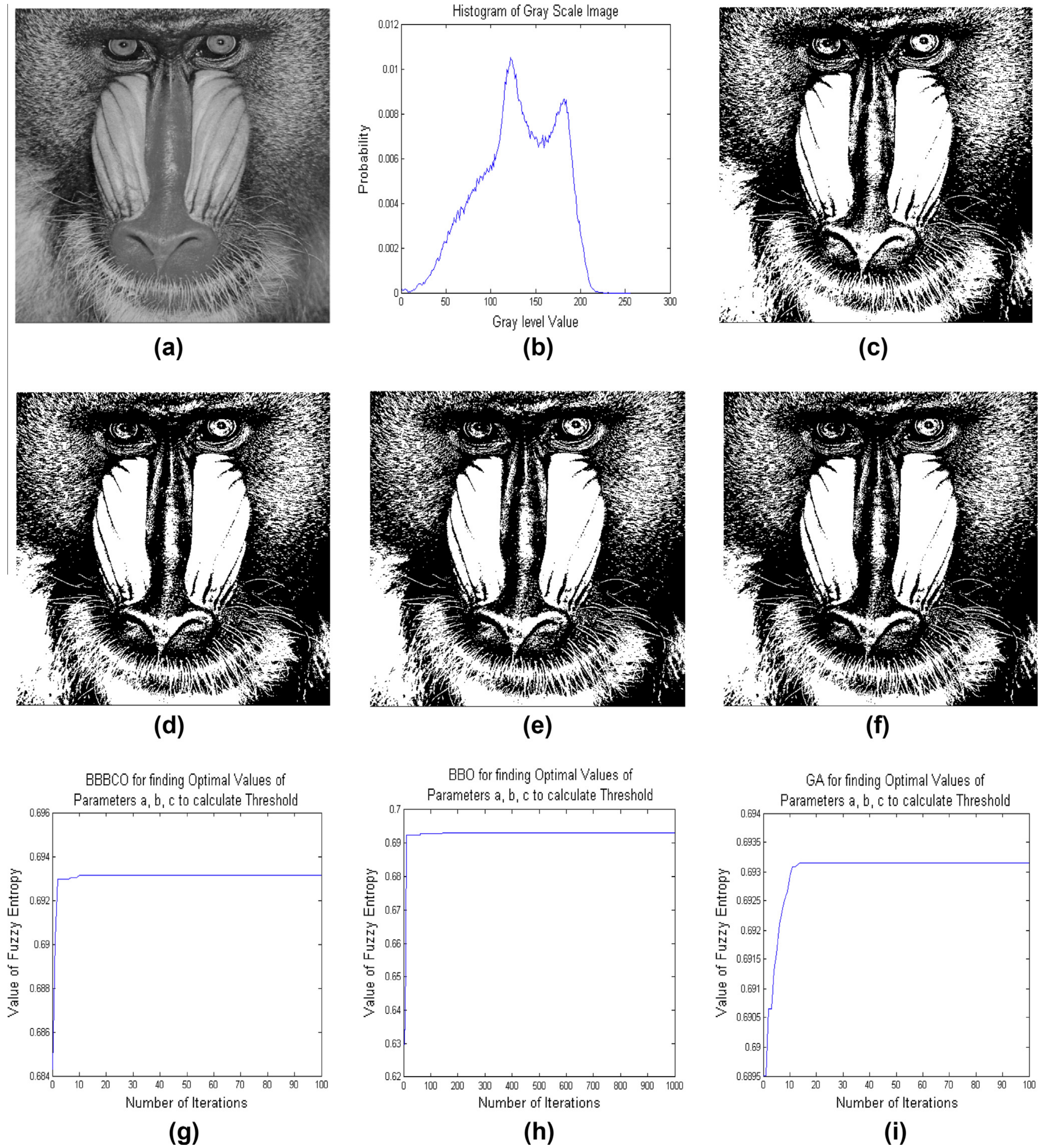

Figure 9 Segmentation of lion image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

for these standard test images. Such optimal values are shown in Table 2.

In order to demonstrate the efficiency of the proposed algorithm, all thresholded results of the proposed algorithm for eleven standard test images are displayed in Figs. 2–12(c). For comparison, all thresholded results of BBO-based fuzzy 2-partition entropy thresholding algorithm, GA-based fuzzy 2-partition entropy thresholding algorithm and recursion-based fuzzy 2-partition entropy thresholding algorithm for different standard test images are displayed in Figs. 2–12(d,e,f),

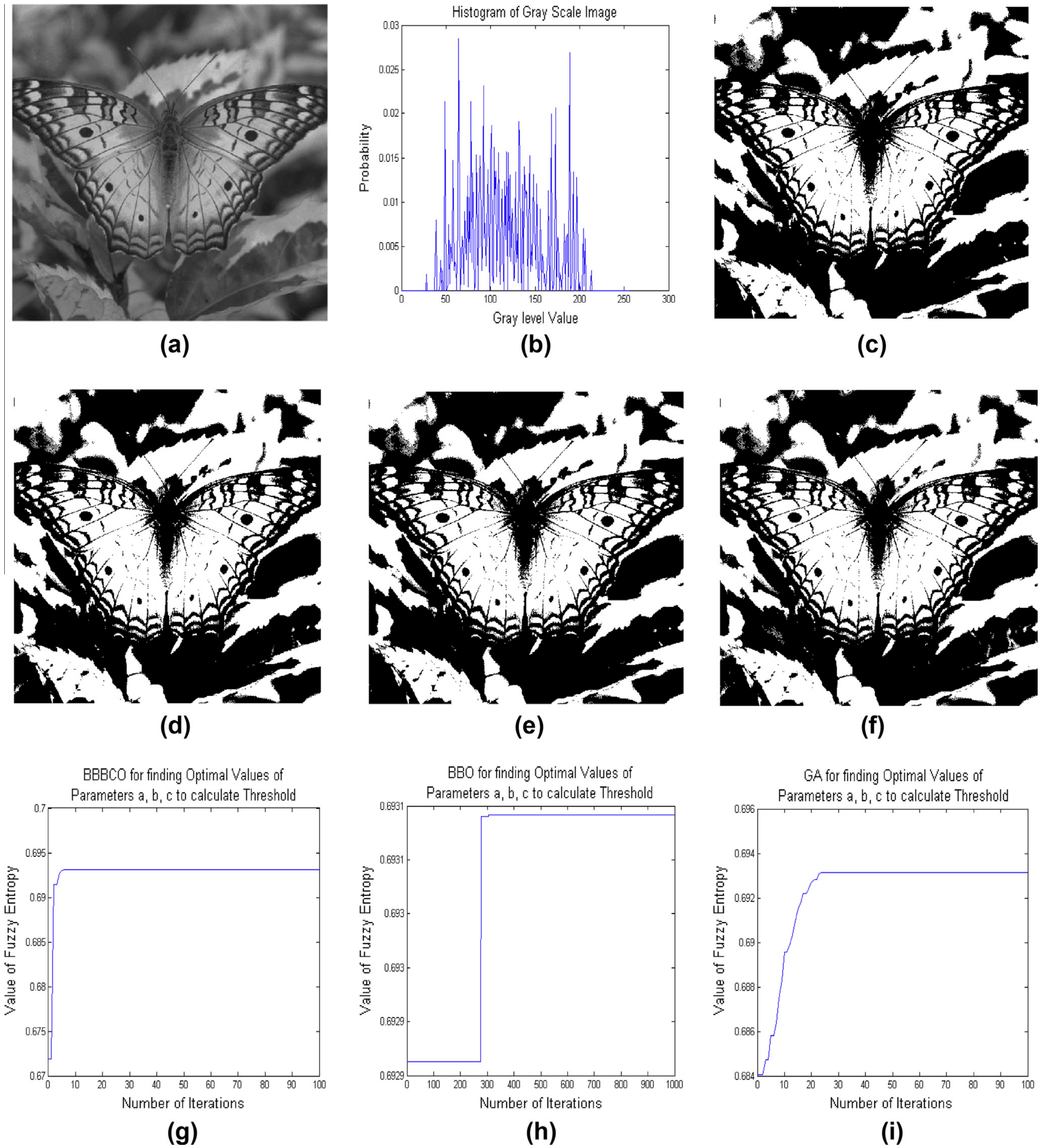

Figure 10 Segmentation of butterfly image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

respectively. From visualization point of view, it is observed that all thresholded results of the proposed algorithm are regarded as acceptable accounting visual effects.

5.2. Performance measures

In practice, however, subjective evaluation of segmented images is usually too inconvenient, time consuming and expensive. So, in this research work, in order to obtain the objective performance of the proposed algorithm and the consistent comparisons, two most popular objective performance measures, peak signal to noise ratio (PSNR) and uniformity, are used. PSNR [10], measured in decibel (dB), is defined as

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{255}{\mathrm{RMSE}}\right) \tag{22}$$

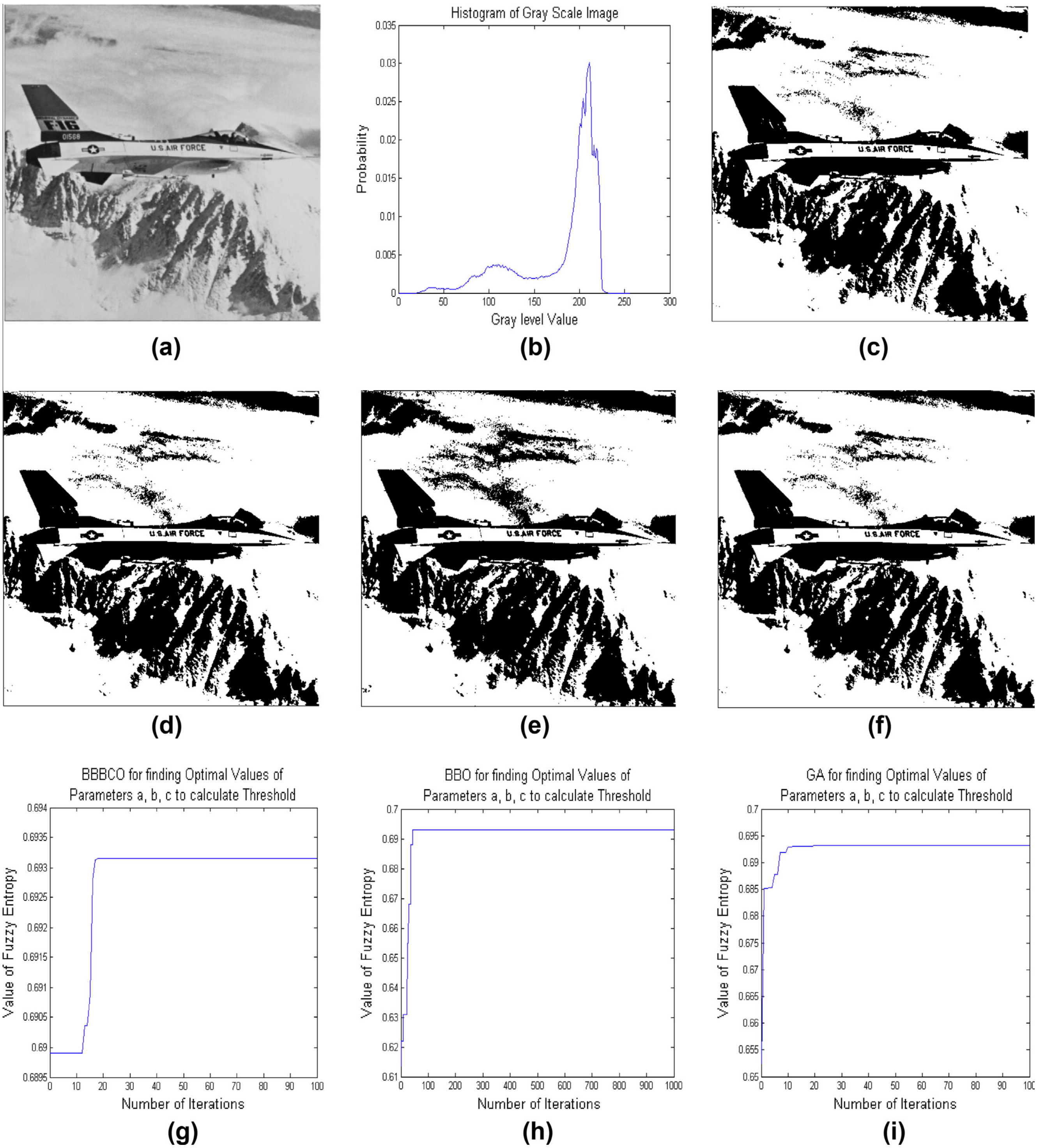

Figure 11 Segmentation of US_airforce_plane image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

where RMSE is the root mean-squared error that is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - S(i,j)]^2}{MN}} \tag{23}$$

Here I and S are input and segmented images of size $M \times N$, respectively.

PSNR is used to determine the quality of the thresholded images. For the better thresholded image quality, the value of PSNR measure should be higher.

Uniformity measure [28] is used to measure region homogeneity in image. It is defined as

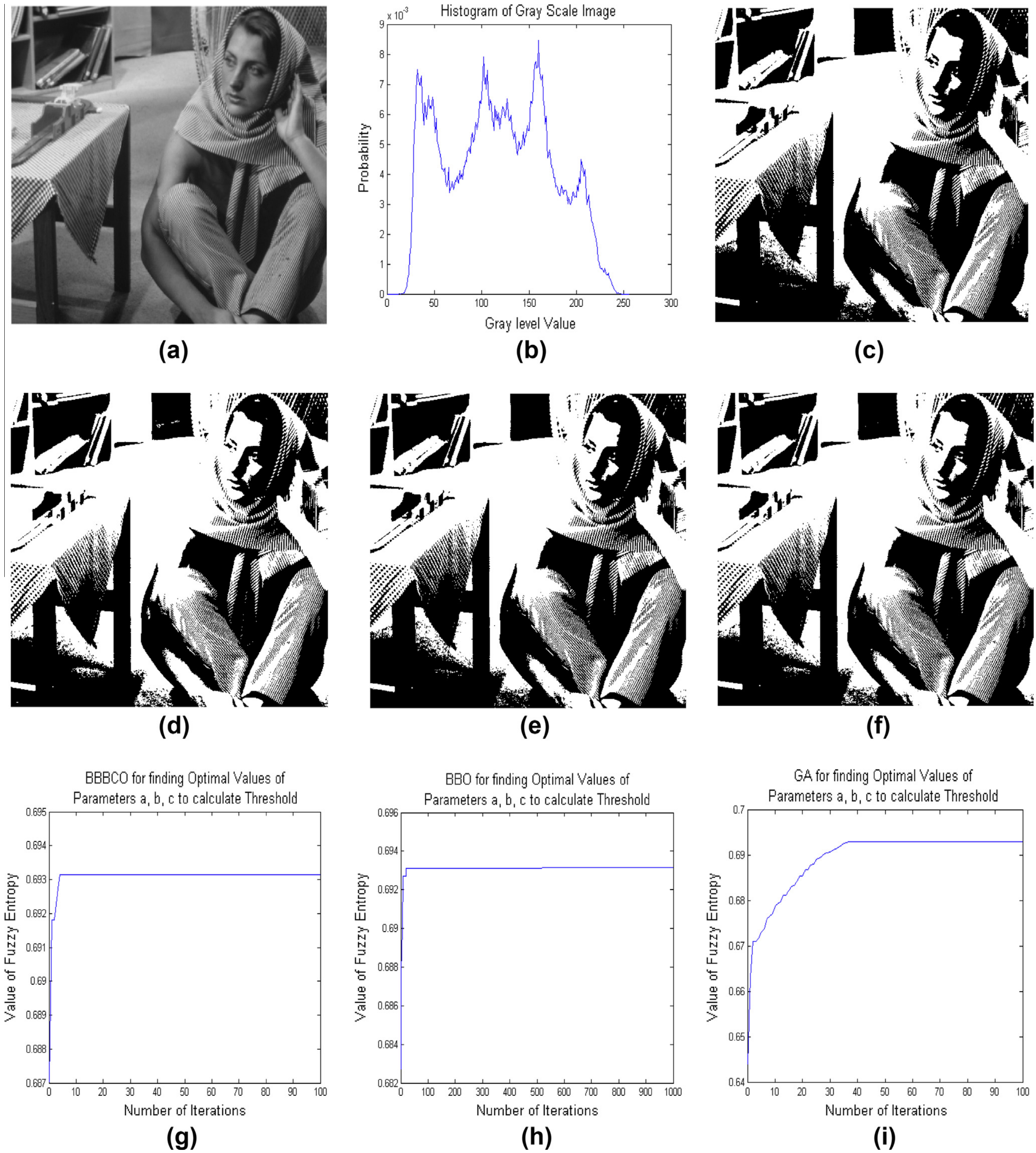

Figure 12 Segmentation of lady image (a) original image, (b) histogram, (c–f) segmentation results of the proposed approach, BBO-based approach, GA-based approach and recursive approach, and (g–i) performance characteristics of BBBCO, BBO and GA to find optimal parameters (a, b, c) that are used for calculating optimal threshold.

$$U = 1 - \frac{\left[2\sum_{k=0}^{1}\sum_{(i,j)\in R_k}\{f(i,j) - \mu_k\}^2\right]}{MN(I_{\max} - I_{\min})^2} \tag{24}$$

where R_k is the kth segmented region; $f(i, j)$ is the gray level value of pixel (i, j); $I_{\max}$ and $I_{\min}$ are the maximum and minimum gray levels in the input image, respectively and μ_k is the mean gray level of pixels in kth region that is defined as

$$\mu_k = \frac{\sum_{(i,j)\in R_k} f(i,j)}{n_k} \tag{25}$$

Here n_k is the total number of pixels in the segmented region R_k.

The value of the uniformity measure lies between 0 and 1. Higher value of uniformity means that the quality of the thresholded image is better [14]. For evaluating the performance of the threshold values obtained according to each algorithm mentioned earlier, PSNR and uniformity measures are used. Table 3 shows the values of PSNR and uniformity measures obtained from four different methods for eleven standard test images. This table provides the values of PSNR and uniformity measures for the comparative study. The bold values of different parameters in table for an algorithm indicate higher performance as compared to other algorithms. From these table values, it is observed that the proposed algorithm is superior to the other three methods.

5.3. Computational complexity analysis

To show the computational complexity analysis of the proposed algorithm, BBO-based fuzzy 2-partition entropy thresholding algorithm, GA-based fuzzy 2-partition entropy

Table 1 Parameters (a, b, c) obtained through four different algorithms for test images.

Image (size)	Proposed algorithm			BBO-based algorithm			GA-based algorithm			Recursive algorithm		
	a	b	c	a	b	c	a	b	c	a	b	c
Cameraman (256 × 256)	50	131	218	23	154	212	56	143	204	30	182	183
Circuit (280 × 272)	56	78	101	55	70	72	0	25	201	22	83	102
Bag (250 × 189)	65	81	116	11	34	187	45	76	86	2	95	137
Glass (181 × 282)	97	121	184	73	124	202	51	151	192	37	149	206
Spine (367 × 490)	0	0	37	3	3	10	0	0	26	1	4	17
Pout (291 × 240)	61	118	147	51	51	246	17	103	212	37	106	190
Moon_surface (256 × 256)	89	90	217	53	161	168	99	133	158	53	146	187
Lion (512 × 512)	103	104	189	77	140	178	61	146	186	1	168	218
Butterfly (512 × 512)	71	132	138	22	113	211	31	121	193	41	88	224
US_airforce_plane (512 × 512)	60	231	250	77	218	254	129	215	227	60	229	255
Lady (512 × 512)	76	128	149	34	91	228	67	121	166	72	101	184

Table 2 Threshold values of four different algorithms for test images.

Image (size)	Proposed algorithm	BBO-based algorithm	GA-based algorithm	Recursive algorithm
Cameraman (256 × 256)	132	134	136	137
Circuit (280 × 272)	78	66	68	71
Bag (250 × 189)	86	70	70	81
Glass (181 × 282)	131	131	134	134
Spine (367 × 490)	10	5	7	6
Pout (291 × 240)	110	108	108	109
Moon_surface (256 × 256)	126	131	130	131
Lion (512 × 512)	128	133	133	135
Butterfly (512 × 512)	116	114	116	112
US_airforce_plane (512 × 512)	187	188	193	188
Lady (512 × 512)	119	112	118	115

Table 3 PSNR and uniformity values resulting from four different methods for test images.

Image (size)	Proposed algorithm		BBO-based algorithm		GA-based algorithm		Recursive algorithm	
	PSNR (dB)	Uniformity	PSNR (dB)	Uniformity	PSNR (dB)	Uniformity	PSNR (dB)	Uniformity
Cameraman (256 × 256)	**9.4814**	**0.9613**	9.4587	0.9595	9.4316	0.9579	9.4141	0.9569
Circuit (280 × 272)	**8.6869**	**0.9698**	7.8710	0.9649	8.0685	0.9664	8.2385	0.9676
Bag (250 × 189)	**10.5192**	**0.9749**	9.8382	0.9707	9.8382	0.9707	10.3399	0.9741
Glass (181 × 282)	**10.8108**	**0.9474**	**10.8108**	**0.9474**	10.7822	0.9456	10.7822	0.9456
Spine (367 × 490)	**4.3540**	**0.9850**	4.1413	0.9788	4.3095	0.9840	4.3095	0.9840
Pout (291 × 240)	**7.5569**	**0.9892**	7.4850	0.9887	7.4850	0.9887	**7.5569**	**0.9892**
Moon_surface (256 × 256)	**7.4212**	**0.9891**	7.3982	0.9885	7.4095	0.9887	7.3982	0.9885
Lion (512 × 512)	**8.5236**	**0.9784**	8.4976	0.9782	8.4976	0.9782	8.4594	0.9779
Butterfly (512 × 512)	**8.5772**	**0.9688**	8.5397	0.9685	**8.5772**	**0.9688**	8.4827	0.9680
US_airforce_plane (512 × 512)	**9.4869**	**0.9744**	9.2356	0.9721	8.4247	0.9648	9.2356	0.9721
Lady (512 × 512)	**9.3682**	**0.9691**	9.2198	0.9688	9.3528	**0.9691**	9.2928	**0.9691**
Mean value	**8.6169**	**0.9734**	**8.4087**	**0.9715**	**8.3797**	**0.9712**	**8.5009**	**0.9721**

Table 4 Number of iterations and computation time of four different algorithms for test images.

Image (size)	Proposed algorithm		BBO-based algorithm		GA-based algorithm		Recursive Algorithm
	Number of iterations	Computation time (s)	Number of iterations	Computation time (s)	Number of iterations	Computation time (s)	Computation time (s)
Cameraman (256 × 256)	26	0.0691	472	1.3209	57	0.1526	1.2572
Circuit (280 × 272)	40	0.1043	531	1.4452	97	0.2619	1.4199
Bag (250 × 189)	13	0.0505	493	2.0008	38	0.1532	1.2539
Glass (181 × 282)	15	0.0568	542	2.0842	37	0.1491	1.2320
Spine (367 × 490)	36	0.1347	624	2.5067	48	0.1852	1.4658
Pout (291 × 240)	8	0.0215	476	1.3450	17	0.0593	1.6178
Moon_surface (256 × 256)	11	0.0367	734	2.4438	16	0.0531	1.2688
Lion (512 × 512)	10	0.2682	689	17.9276	15	0.3906	1.8522
Butterfly (512 × 512)	10	0.2352	306	7.2409	25	0.5948	1.9255
US_airforce_plane (512 × 512)	18	0.4934	734	19.8297	25	0.6774	1.8272
Lady (512 × 512)	10	0.2527	520	13.1934	42	1.0091	1.8554
Mean Value	18	0.1566	556	6.4853	38	0.3351	1.5432

thresholding algorithm and recursion-based fuzzy 2-partition entropy thresholding algorithm, the processing time of the proposed algorithm is compared with those of BBO-based fuzzy 2-partition entropy thresholding algorithm, GA-based fuzzy 2-partition entropy thresholding algorithm and recursion-based fuzzy 2-partition entropy thresholding algorithm. The proposed algorithm, BBO-based fuzzy 2-partition entropy thresholding algorithm and GA-based fuzzy 2-partition entropy thresholding algorithm are iterative algorithms. Thus, the proposed algorithm is also compared with BBO-based fuzzy 2-partition entropy thresholding algorithm and GA-based fuzzy 2-partition entropy thresholding algorithm based on the number of iterations required to find the optimal values of parameters a, b and c. The performance characteristics of iterative methods: BBBCO-based, BBO-based and GA-based are displayed in Figs. 2–12(g,h,i). Table 4 shows the number of iterations performed by iterative algorithms and the computation time of the four different methods. From the table values, it is found that the proposed algorithm is faster than the other three algorithms.

6. Conclusion and future work

This paper presents a new fuzzy 2-partition entropy thresholding approach based on the Big Bang–Big Crunch Optimization algorithm. The proposed algorithm is compared with three different algorithms included BBO-based fuzzy 2-partition entropy thresholding algorithm, GA-based fuzzy 2-partition entropy thresholding algorithm and recursion-based fuzzy 2-partition entropy thresholding algorithm. For evaluating the performance of the proposed algorithm, experiments are conducted on a number of standard test images. To check the effectiveness of the proposed method, three performance metrics are used. These performance metrics are PSNR, uniformity and computation time. PSNR and uniformity are used to measure the quality of the thresholded images while computation time is used to get an idea of complexity. From the experimental results based on PSNR and uniformity, it is observed that the performance of the proposed algorithm is better than other three algorithms. From computational point of view, the proposed algorithm is faster than other three algorithms. Thus, the experimental results of the proposed approach are quite promising. These results encourage further research for applying BBBCO-based fuzzy 2-partition entropy thresholding algorithm on medical images to develop computer-aided disease diagnosis systems.

References

[1] Quintanilla-Dominguez J, Ojeda-Magana B, Cortina-Januchs MG, Ruelas R, Vega-Corona A, Andina D. Image segmentation by fuzzy and possibilistic clustering algorithms for the identification of microcalcifications. Sci Iranica 2011;18(3):580–9.

[2] Shitong Wang, Chung FL. Note on the equivalence relationship between Renyi-entropy and Tsallis-entropy based thresholding. Pattern Recogn Lett 2005;26(14):2309–12.

[3] Pun T. Entropies thresholding, a new approach. Comput Vision, Graphics Image Process 1981;16(2):210–39.

[4] Kapur JN. A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vision, Graphics Image Process 1985;29(3):273–85.

[5] Sahoo Prasanna, Wilkins Carrye, Yeager Jerry. Threshold selection using Renyi's entropy. Pattern Recogn 1997;30(1):71–84.

[6] de Albuquerque M Portes, Esquef IA, Gesualdi Mello AR. Image thresholding using Tsallis-entropy. Pattern Recogn Lett 2004;25(9):1059–65.

[7] Li CH, Lee CK. Minimum cross entropy thresholding. Pattern Recogn 1993;26(4):617–25.

[8] Nie Fangyan, Gao Chao, Guo Yongcai, Gan Min. Two-dimensional minimum local cross-entropy thresholding based on co-occurrence matrix. Comput Electr Eng 2011;37(5):757–67.

[9] Tang Kezong, Yuan Xiaojing, Sun Tingkai, Yang Jingyu, Gao Shang. An improved scheme for minimum cross entropy threshold selection based on genetic algorithm. Knowl-Based Syst 2011;24(8):1131–8.

[10] Horng Ming-Huwi. Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst Appl 2010;37(6):4580–92.

[11] Horng Ming-Huwi, Liou Ren-Jean. Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Syst Appl 2011;38(12):14805–11.

[12] Cheng HD, Cheng Jim-Rong Chen, Li Jiguang. Threshold selection based on fuzzy c-partition entropy approach. Pattern Recogn 1998;31(7):857–70.

[13] Benabdelkader Souad, Boulemden M. Recursive algorithm based on fuzzy 2-partition entropy for 2-level image thresholding. Pattern Recogn 2005;38(8):1289–94.

[14] Tang Yinggan, Weiwei Mu, Zhang Ying, Zhang Xuguang. A fast recursive algorithm based on fuzzy 2-partition entropy approach for threshold selection. Neurocomputing 2011;74(8):3072–8.

[15] Tao Wen-Bing, Tian Jin-Wen, Liu Jian. Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. Pattern Recogn Lett 2003;24(16):3069–78.

[16] Tao Wenbing, Jin Hai, Liu Liman. Object segmentation using ant colony algorithm and fuzzy entropy. Pattern Recogn Lett 2007;28(7):788–96.

[17] Chatterjee Amitava, Siarry Patrick, Nakib Amir, Blanc Raphael. An improved biogeography based optimization approach for segmentation of human head CT-scan images employing fuzzy entropy. Eng Appl Artif Intell 2012;25(8):1698–709.

[18] Sanyal Nandita, Chatterjee Amitava, Munshi Sugata. An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. Expert Syst Appl 2011;38(12):15489–98.

[19] Erol Osman K, Eksin Ibrahim. A new optimization method: big bang–big crunch. Adv Eng Softw 2006;37(2):106–11.

[20] Tang Hesheng, Zhou Jin, Xue Songtao, Xie Liyu. Big bang–big crunch optimization for parameter estimation in structural systems. Mech Syst Signal Process 2010;24(8):2888–97.

[21] Hasancebi O, Kazemzadeh Azad S. An exponential big bang–big crunch algorithm for discrete design optimization of steel frames. Comput Struct 2012;110–111:167–79.

[22] Deihimi Ali, Solat Alireza. Optimized echo state networks using a big bang–big crunch algorithm for distance protection of series-compensated transmission lines. Electr Power Energy Syst 2014;54:408–24.

[23] Kumbasar Tufan, Hagras Hani. Big bang–big crunch optimization based interval type-2 fuzzy PID cascade controller design strategy. Inf Sci 2014;282:277–95.

[24] Yesil Engin. Interval type-2 fuzzy PID load frequency controller using big bang–big crunch optimization. Appl Soft Comput 2014;15:100–12.

[25] Cheng HD, Xu Huijuan. Fuzzy approach to contrast enhancement. In: Proceedings of IEEE 14th international conference on pattern recognition; 1998. p. 1549–51.

[26] Khehra Baljit Singh, Pharwaha Amar Partap Singh. Automatic detection of microcalcifications in digitized mammograms using fuzzy 2-partition entropy and mathematical morphology. In: Proceedings of the IASTED international conference on signal and image processing; 2012. p. 220–227.

[27] Shannon CE. A mathematical theory of communication. Int J Bell Syst Tech 1948;27:379–423.

[28] Yin Shibai, Zhao Xiangmo, Wang Weixing, Gong Minglun. Efficient multilevel image segmentation through fuzzy entropy maximization and graph cut optimization. Pattern Recogn 2014;47:2894–907.