

# Grayscale Image Segmentation Using Multilevel Thresholding and Nature-Inspired Algorithms

Genyun Sun, Aizhu Zhang and Zhenjie Wang

**Abstract** Multilevel image thresholding plays a crucial role in analyzing and interpreting the digital images. Previous studies revealed that classical exhaustive search techniques are time consuming as the number of thresholds increased. To solve the problem, many nature-inspired algorithms (NAs) which can produce high-quality solutions in reasonable time have been utilized for multilevel thresholding. This chapter discusses three typical kinds of NAs and their hybridizations in solving multilevel image thresholding. Accordingly, a novel hybrid algorithm of gravitational search algorithm (GSA) with genetic algorithm (GA), named GSA-GA, is proposed to explore optimal threshold values efficiently. The chosen objective functions in this chapter are Kapur's entropy and Otsu criteria. This chapter conducted experiments on two well-known test images and two real satellite images using various numbers of thresholds to evaluate the performance of different NAs.

**Keywords** Image segmentation · Multilevel thresholding · Nature-inspired algorithms (NAs) · Gravitational search algorithm (GSA) · Genetic algorithm (GA) · Kapur's entropy · Otsu

## 1 Introduction

For the analysis, interpretation, and understanding of digital images, image segmentation is one of the most essential and fundamental technologies [1]. It is useful for partitioning a digital image into multiple regions/objects with distinct gray-levels [2]. Over the several decades, three main categories in image segmentation have been

---

G. Sun (✉) · A. Zhang · Z. Wang  
School of Geosciences, China University of Petroleum,  
Qingdao 266580, Shandong, China  
e-mail: genyunsun@163.com

A. Zhang  
e-mail: zhangaizhu789@163.com

Z. Wang  
e-mail: sdwj@upc.edu.cn

exploited in the literatures: (1) edge-based image segmentation, (2) region-based image segmentation, and (3) special theory-based image segmentation. Detailed introduction of their properties are presented in [3]. Particularly, thresholding technique of the region-based segmentation is regard as the most popular one out of others [4]. The main purpose of image thresholding is to determine one (bi-level thresholding) or  $m$  (multilevel thresholding) appropriate threshold values for an image to divide pixels of the image into different groups [5, 6]. In the recent years, increasing complexity of digital images, such as intensity inhomogeneity, makes multilevel thresholding approaches drawn much more attention. This is mainly due to its easy implementation and low storage memory characteristic [7].

Generally, the multilevel thresholding transforms the image segmentation to an optimization problem where the appropriate threshold values are found by maximizing or minimizing a criterion. For example, in the popular Otsu's method [8], thresholds are determined by maximizing the between-class variance. In the Kapur's entropy [9], the optimum thresholds are achieved by maximizing the entropy of different classes. A fuzzy entropy measure is applied for picking the optimum thresholds in [10] while Qiao et al. [11] formulated the thresholding criterion by exploring the knowledge in terms of intensity contrast. Researches have also developed some other preferable criteria, including fuzzy similarity measure [12], Bayesian error [13], cross entropy [14], Tsallis entropy [15], and so on.

Exhaustive search algorithms achieved optimization of the aforementioned criteria. However, the methods will become quiet time consuming if the number of desired thresholds is increased [16]. Moreover, the exhaustive search algorithms prone to premature convergence when dealing with complex real life images [17–19]. Nature-inspired algorithms (NAs) possess the ability for searching the optimal solutions on the basis of any objective function [7]. Furthermore, the population-based nature of NAs allows the generation of several candidate thresholds in a single run [20]. The population-based nature of NAs thus remarkably reduces the computational time of multilevel thresholding. Consequently, many NAs have been preferred in finding optimum thresholds for image thresholding.

The NAs can be categorized into three typical kinds: swarm intelligence algorithms (SIAs), evolutionary algorithms (EAs), and physics-based algorithms (PAs) in accordance with their original inspirations. Popular NAs chosen for multilevel thresholding include genetic algorithm (GA) [8, 21–24], differential evolution (DE) [18, 25–27], simulated annealing (SA) [16, 28], ant colony optimization (ACO) [29, 30], artificial bee colony optimization (ABC) [17, 31], differential search algorithm (DS) [16, 32], particle swarm optimization (PSO) [17, 33–35], and so on. Generally speaking, all these algorithms have achieved certain successes and have showed different advantages. For example, DE achieves higher quality of the optimal thresholds than GA, PSO, ACO, and SA whereas PSO converges the most quickly when comparing with ACO, GA, DE, and SA [21]. Besides, the DS consumes the shortest running time for multilevel color image thresholding when comparing with DE, GA, PSO, ABC, etc. [16].

Moreover, to further improve the optimization ability of these NAs, large amounts of variants, such as hybrid algorithms have been proposed. Many of the hybrid

algorithms have been utilized in solving multilevel thresholding problems, including GAPSO (hybrid GA with PSO) [36, 37], ACO/PSO (hybrid ACO with PSO) [38], SA/PSO (hybrid SA with PSO) [39], BBO-DE (hybrid DE with biogeography-based optimization (BBO)) [40] etc. It is demonstrated that these hybrid algorithms can always perform more effective search ability than the original ones. But simple hybrid model easily causes high-computational complexity.

Therefore, in this chapter, we developed a novel hybrid algorithm of GSA with GA (GSA-GA) for multilevel thresholding. The proposed GSA-GA is expected to rapidly obtain the high-quality optimal thresholds. In GSA-GA, the discrete mutation operator [41] is introduced to promote the population diversity when premature convergence occurred. Moreover, for selecting the particles for mutation, the roulette selection [42] is also introduced. The introduction of these operators therefore could promote GSA-GA to perform faster and more accurate multilevel image thresholding. Both Kapur's entropy and Otsu are considered as evaluation criteria for GSA-GA.

Section 2 formulates the multilevel thresholding problem as an optimization problem first and then introduces two criteria briefly. In Sect. 3, typical NAs and hybrid NAs based multilevel image thresholding are overviewed. Section 4 describes the proposed GSA-GA algorithm and states its application in multilevel thresholding. In Sect. 5, experiments are conducted over two standard test images and two real satellite images to evaluate the effectiveness of the GSA-GA. Finally, the chapter is concluded in Sect. 6.

## 2 Formulation of Multilevel Thresholding Problem

Optimal thresholding methods search for the optimum thresholds by minimizing or maximizing an objective function. Parameters in the utilized objective function are made up by the selected thresholds in each iteration. Specially, the purpose of multilevel thresholding is to classify the  $Num$  pixels of an image into  $K$  classes using a set of thresholds  $Th = (td_1, td_2, \dots, td_m)$  where  $m = K - 1$ . Obviously, different sets of thresholds will produce different image segmentation results. Researchers have formulated some criteria for constructing efficient objective functions [8, 9, 12, 13, 43, 44]. Two most preferred criteria are Kapur's entropy and Otsu methods as they are relatively easy to use. They are also chosen as criteria in this chapter.

### 2.1 Kapur's Entropy Criterion

The Kapur's entropy criterion, proposed by Kapur et al. [9], is originally developed for bi-level image thresholding. In this method, proper segmentation of an image is achieved by maximizing the Kapur's entropy. For a bi-level thresholding, the Kapur's entropy can be described as follows:

$$\begin{aligned}
HE_0 &= -\sum_{i=0}^{td_0-1} \frac{prob_i}{\omega_0} \ln \frac{prob_i}{\omega_0}, \omega_0 = \sum_{i=0}^{td_0-1} prob_i \\
HE_1 &= -\sum_{i=td_0}^{L-1} \frac{prob_i}{\omega_1} \ln \frac{prob_i}{\omega_1}, \omega_1 = \sum_{i=td_0}^{L-1} prob_i
\end{aligned} \tag{1}$$

where  $HE_i$  is the entropy of class  $i$  ( $i \in 0, 1$ ) and  $\omega_i$  is the probability of class  $i$ ,  $td_0$  is the optimum threshold of the bi-level thresholding. Assume that there are  $L$  gray-levels in a given image, the probability of the gray-level  $i$  can be defined as follows:

$$prob_i = \frac{hist_i}{\sum_{i=0}^{L-1} hist_i}, \tag{2}$$

where  $hist_i$  denotes the number of pixels with gray-level  $i$ .

As shown in (1) and (2),  $td_0$  is the selected threshold. To obtain the optimum threshold  $td_0$ , the fitness function (3) needs to be maximized as follows:

$$f(td_0) = HE_0 + HE_1, \tag{3}$$

To solve complex multilevel thresholding problems, the Kapur's entropy has been extended to determine  $m$  thresholds for a given image, i.e.,  $[td_1, td_2, \dots, td_m]$ . The objective of the Kapur's entropy-based multilevel thresholding is to maximize the fitness function (4) as follows:

$$f([td_1, td_2, \dots, td_m]) = HE_0 + HE_1 + HE_2 + \dots + HE_m, \tag{4}$$

where

$$\begin{aligned}
HE_0 &= -\sum_{i=0}^{td_1-1} \frac{prob_i}{\omega_0} \ln \frac{prob_i}{\omega_0}, \omega_0 = \sum_{i=0}^{td_1-1} prob_i \\
HE_1 &= -\sum_{i=td_1}^{td_2-1} \frac{prob_i}{\omega_1} \ln \frac{prob_i}{\omega_1}, \omega_1 = \sum_{i=td_1}^{td_2-1} prob_i \\
HE_2 &= -\sum_{i=td_2}^{td_3-1} \frac{prob_i}{\omega_2} \ln \frac{prob_i}{\omega_2}, \omega_2 = \sum_{i=td_2}^{td_3-1} prob_i \\
HE_m &= -\sum_{i=td_m}^{L-1} \frac{prob_i}{\omega_m} \ln \frac{prob_i}{\omega_m}, \omega_m = \sum_{i=td_m}^{L-1} prob_i.
\end{aligned} \tag{5}$$

## 2.2 Otsu Criterion

The Otsu criterion [8] is another histogram-based multilevel image thresholding algorithm. The method divides the given image into  $m$  classes so that the total variance of different classes is maximized. The object function of Otsu is defined as the sum of sigma functions of each class as follows:

$$f(td_0) = \sigma_0 + \sigma_1, \tag{6}$$

The sigma functions are as follows:

$$\begin{aligned}\sigma_0 &= \omega_0(\mu_0 - \mu_T)^2, \mu_0 = \frac{\sum_{i=0}^{id_0-1} i \cdot prob_i}{\omega_0}, \\ \sigma_1 &= \omega_1(\mu_1 - \mu_T)^2, \mu_1 = \frac{\sum_{i=id_0}^{L-1} i \cdot prob_i}{\omega_1},\end{aligned}\quad (7)$$

where  $\mu_i$  is the mean gray-level of class  $i$  ( $i \in 0, 1$ ) and  $\mu_T$  is the mean intensity of the original image.

Similar to the entropy method, the Otsu criterion has already been extent to multi-level thresholding problems. The objective in the Otsu-based multilevel thresholding is to maximize the following function:

$$f([td_1, td_2, \dots, td_m]) = \sigma_0 + \sigma_1 + \sigma_2 + \dots + \sigma_m, \quad (8)$$

where

$$\begin{aligned}\sigma_0 &= \omega_0(\mu_0 - \mu_T)^2, \mu_0 = \frac{\sum_{i=0}^{id_1-1} i \cdot prob_i}{\omega_0}, \\ \sigma_1 &= \omega_1(\mu_1 - \mu_T)^2, \mu_1 = \frac{\sum_{i=id_1}^{id_2-1} i \cdot prob_i}{\omega_1}, \\ \sigma_2 &= \omega_2(\mu_2 - \mu_T)^2, \mu_2 = \frac{\sum_{i=id_2}^{id_3-1} i \cdot prob_i}{\omega_2}, \\ \sigma_m &= \omega_m(\mu_m - \mu_T)^2, \mu_m = \frac{\sum_{i=id_m}^{L-1} i \cdot prob_i}{\omega_m}.\end{aligned}\quad (9)$$

### 3 Nature-Inspired Algorithms Based Multilevel Thresholding

In this chapter, we categorized the NAs utilized for multilevel thresholding into three kinds: SIAs, EAs, and PAs based on their original inspirations. Since different kinds of NAs possess unique information sharing mechanisms, certain successes have been produced by each kind of NA. Sections 3.1–3.3 give a detailed review of the three aforementioned kinds of NAs based multilevel thresholding, respectively. In Sect. 3.4, we briefly reviewed the hybrid NAs based multilevel thresholding.

#### 3.1 Swarm Intelligence-Based Optimization Algorithms Based Multilevel Thresholding

SIAs are mainly proposed by mimicking the foraging or hunting behaviors of different species like ants, bees, birds, cuckoos, bats, wolf, and so on [45]. Well-known SIAs include ACO [29], ABC [31], PSO [46], cuckoo search algorithm (CS) [47], Bat Algorithm (BA) [48], and Gray Wolf Optimizer (GWO) [49], etc. In these algorithms, one special particle/individual with the best performance is usually chosen

as exemplar/leader for others. The guidance of the exemplar ensures the population evolves directionally in each iteration. The directional search property equips the SIAs with the fast convergence performance [50].

Application of swarm intelligence thereby makes a success in decreasing the computational complexity problem of multilevel thresholding. For example, in [30], Tao et al. utilized ACO to optimize the fuzzy entropy criterion while Zhang et al. search for the optimum multilevel thresholding on the basis of the maximum Tsallis entropy [51]. Similarly, ABC, BA, and GWO also obtained certain successes in multilevel thresholding [6, 17, 51–54]. Besides, as one of the most popular swarm intelligence-based optimization algorithm, PSO and its various variants have contributed to multilevel thresholding a lot [2, 17, 33–35, 55]. Moreover, Hammouche et al. have found that PSO converges more quickly than ACO, ABC, and SA because of all particles can directly learn from the global best particle [21].

Although SIAs contribute a lot to image thresholding, their fast convergence property caused rapid decrease of population diversity. That is, exploration of SIAs is insufficient. This defective exploration of SIAs may result in temperature convergence of multilevel thresholding.

### ***3.2 Evolutionary Algorithms Based Multilevel Thresholding***

EAs are motivated by the principle of evolution through selection and mutation. In the past decade, many EAs, such as GA [8], DE [27], and DS [32], have been widely applied. In conventional EAs, evolutionary processes are conducted through reproduction, crossover, mutation, and selection operators. Particularly, only two individuals are randomly selected to exchange information in a single crossover operation [50]. Similarly, in a traditional mutation operator, only two genes in the individuals are randomly varied [50]. Obviously, evolutionary performance of EAs is not as directional as SIAs do. It is omnidirectional to some extent. Thereby EAs have better exploration than SIAs.

Naturally, a number of researchers have paid close attention to obtain optimal thresholds by EAs, especially the GA [21–24] and DE [18, 25, 26]. The favorable exploration ability of EAs makes them achieve remarkable progress in the research area of multilevel thresholding. However, the omnidirectional characteristic of EAs leads to their slow convergence. Moreover, the success of some EAs in solving specific problems highly rely on the user set trial vector generation strategies and control parameters [16].

### ***3.3 Physics-Based Algorithms Based Multilevel Thresholding***

The PAs, as the name suggested, are constructed by simulating the phenomenon of physics. For instance, SA [28] imitates the annealing process of melted iron;

gravitational search algorithm (GSA) [56] inspired by the gravitational kinematics; electromagnetism-like algorithm (EM) [57] and charged search system (CSS) algorithm [58] are proposed on the basis of electrodynamics. More detailed survey of the PAs can be found in [59]. Individuals in PAs explore the search space follows several exemplars. Accordingly, interactions among individuals result in iterative improvement of solutions quality over time. Search diversity of PAs is better than that of SIAs and thus PAs usually perform better exploration. On the other hand, their more directional search property keep higher convergence speed of them. Nevertheless, none of the PAs can solve all the optimization problems in accordance with the “No Free Lunch Theorems” [60].

Comparing with the SIAs and EAs, few researchers have focused on their application on multilevel thresholding. GSA, which possesses simple concept, high-computational efficiency, and few parameters, has proven its promising efficiency in solving complex problems [56, 61]. Furthermore, numerical experiments has confirmed the superiority of GSA with respect to search ability and speed over many other NAs, such as PSO, GA, ACO etc. [56, 62–64]. These advantages make GSA a potential choice for solving multilevel thresholding.

### ***3.4 Hybrid NAs Based Multilevel Thresholding***

As discussed in Sects. 3.1–3.3, each kind of NAs has separate and unique advantages and disadvantages. To incorporate different advantages of various NAs and improve the optimization ability of them, large amounts of hybrid algorithms have been proposed over the past decades. For instance, in GAPSO [9], the mutation operator of GA is introduced to PSO to tackle the premature convergence problem while the memory property of PSO is preserved. BBO-DE [40] incorporate both the crossover operator and the selection operator of DE algorithm to BBO to promote exploration ability of BBO while the exploitation ability of BBO is kept. Experiments reported in the literatures demonstrated that these hybrid algorithms can always perform more effective search ability than the original ones.

Many of the hybrid algorithms have already been utilized to solving multilevel thresholding problems, including GAPSO [9], ACO/PSO [38], SA/PSO [39], and BBO-DE [40], etc. The hybrid algorithms usually possess advantages of each composed algorithm. Due to simple hybrid model easily causes high-computational complexity, new NAs, or efficient hybrid NAs are desirable.

## **4 GSA-GA Based Multilevel Image Thresholding**

This chapter introduces a novel hybrid GSA-GA for multilevel image thresholding [65]. Although many researches have paid close attention to the improvement of GSA and presented some GSA variants in the past few years [1, 61, 66, 67], very

few of the algorithms have focused on the application on multilevel thresholding. When applying GSA into multilevel image thresholding, especially when the desired number of thresholds is large, the premature convergence and high time-consuming problems become more serious.

The lack of diversity is one important reason for the premature convergence [68]. In GSA-GA, the discrete mutation operator of GA was introduced to promote the population diversity when premature convergence occurred. Moreover, for decreasing computational time, adaptive method is utilized to judge whether the hybridization of GSA and GA are performed. The details of GSA-GA and its application for multilevel image thresholding are given in Sects. 4.1 and 4.2, respectively.

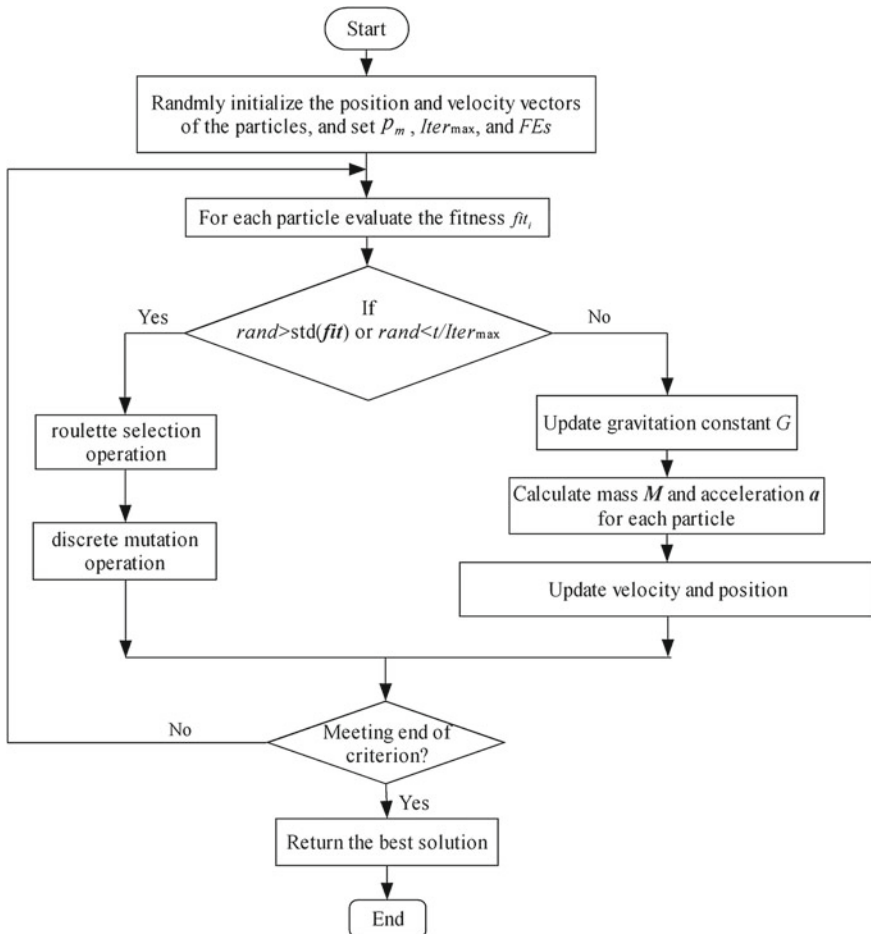


Fig. 1 The GSA-GA principle



## 4.1 GSA-GA

In GSA-GA, to adaptively identify whether the hybridization of GSA and GA is necessary, the standard deviations of objective functions and the ratio of the current iteration  $t$  to the maximum iterations  $maxiterations$  are calculated before conducting the selection and mutation operators of GA. More specifically, when particles are trapped (judging by  $rand < std(fit)$  or  $rand < t/maxiterations$ ), the roulette selection and discrete mutation operators are carried out to update position of each particle and thus to diversify the population. Due to the population diversity is crucial for a

```

1. Input test image IMG.
2. Set  $P_m$ ,  $Iter_{max}$ ,  $N$ ,  $G_0$ ,  $\beta$ ,  $m$ , and  $FES$ .
3. Set the thresholding criterion  $F_{index} = 'entropy'$  or 'between-class variance'.
4. Initialize positions  $x$  and velocities  $v$  of particles in the population.
5. For  $t=1:Iter_{max}$ 
6.   If  $F_{index} = 'entropy'$ 
7.     Evaluate the fitness functions values  $fit$  by Eq. (4);
8.   Else
9.     Evaluate the fitness functions values  $fit$  by Eq. (8);
10.  End
11. End
12. If ( $rand > std(fit) \parallel rand < (t/Iter_{max})$ )
13.   Calculate the selection probability  $p'_s(i)$  of each particle  $i$ ;
14.   Calculate cumulative probability  $p'_{cs}(i)$  of each particle  $i$ ;
15.   For  $i=1:N$ 
16.     Generate  $rand$  randomly;
17.     If  $p'_{cs}(j) < rand < p'_{cs}(j+1)$   $j \in [1, 2, \dots, N]$ 
18.        $Newx'_j = x'_j$  ;
19.     End
20.   End
21.   Randomly generate basic vector BaseV for discrete mutation in the gray range of IMG ;
22.   Create mutation mask matrix by  $BaseM = BaseV(ones(N, I);)$  ;
23.    $x'^{t+1} = rem(Newx' + (rand < p_m) .* ceil(rand .* (BaseM - 1)), BaseM)$  ;
24. Else
25.   Update the gravitation constant by  $G(t) = G_0 \times \exp(-\beta \times \frac{t}{Iter_{max}})$  ;
26.   For  $i=1:N$ 
27.     Calculate the mass by  $Mass'_i = \frac{nmfit'_i}{\sum_{j=1}^N nmfit'_j}$  and  $nmfit'_i = \frac{fit'_i - worst^t}{best^t - worst^t}$  ;
28.     Calculate the force acting on particle  $i$  from other particles in demension  $d$ 
29.     by  $F_{ij}^d(t) = G(t) \frac{Mass'_i \times Mass'_j}{R_{ij}^d + \epsilon} (x'_{jd} - x'_{id})$  and  $F_i^d(t) = \sum_{j=1, j \neq i}^N rand \cdot F_{ij}^d(t)$  ;
30.     Calculate the acceleration by  $a'_{id} = \frac{F_i^d(t)}{Mass'_i}$ ;
31.     Update the position and velocity of particle  $i$  by:
32.      $v'^{t+1}_{id} = rand \times v'_{id} + a'_{id}$  ;
33.      $x'^{t+1}_{id} = x'^t_{id} + v'^{t+1}_{id}$  ;
34.   End
35. End
36. End

```

**Fig. 2** Pseudocode of the GSA-GA algorithm for multilevel image thresholding

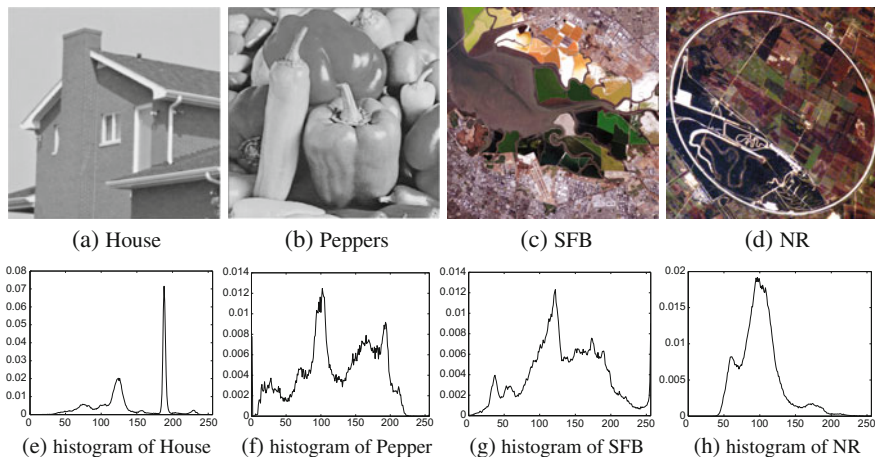
health search, combination of GA, and GSA can help the algorithm escape from local optima. Besides, because of satisfaction of the condition ' $rand < t/maxiterations$ ' becomes easier by the lapse of time, GSA-GA can utilize selection and mutation operators to accelerate convergence in the last iterations. The principle of GSA-GA is shown in Fig. 1. The detailed introduction of the method can be found in [65].

## 4.2 Implementation of GSA-GA for Multilevel Thresholding

The application of GSA-GA approach to the multilevel image thresholding problem depends on the criterion used for optimization. In this chapter, the Kapur's entropy and Otsu criteria were implemented. To start the GSA-GA for multilevel thresholding, initial population  $X = [X_1, X_2, \dots, X_i, \dots, X_N]$  where  $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$  should be randomly generated first. Each particle  $X_i$  is comprised of a set of gray values, which stands for a candidate solution of the required threshold values. The size of the population is  $N$  (can be set by users), and dimension of each particle is  $D$ , which is also the number of desired thresholds:  $D = m$ . In the iteration process, the fitness value of each particle is calculated from the Kapur's entropy or Otsu criterion using Eqs. (4)–(5) and (8)–(9), respectively. The pseudocode of the GSA-GA algorithm for multilevel image thresholding is shown in Fig. 2.

## 5 Experimental Results and Discussion

In this section, the experimental results are presented. The performance of GSA-GA has been evaluated and compared with the other 6 NAs: the standard GSA [56], GGSA (Adaptive gbest-guided GSA) [67], PSO-GSA (hybrid PSO and GSA) [66], DS (Differential Search algorithm) [32], BBO-DE [40], and GAPSO [36] by conducting experiments on two benchmark images that has been tested in many related literatures. Furthermore, to evaluate the ability of GSA-GA based thresholding on more complex and difficulty images, we conducted experiments on two real satellite images adopted from NASA (<http://earthobservatory.nasa.gov/Images/?eocn=topnav046eoci=images>). The obtained experimental results of the 7 algorithms are also compared. For the satellite images, the low spectral and spatial resolution usually makes objects hard to be segmented, while the high resolution always leads to highly computational complexity during the image segmentation. The four experimental images and corresponding histogram of each image are given in Fig. 3 as shown in Sect. 5.1.



**Fig. 3** Test images used in the experiments

## 5.1 Test Images

The two standard benchmark test images are two widely utilized images: House and pepper, as shown in Fig. 3a and b, respectively. Size of every tested benchmark images is  $256 \times 256$ . The two tested satellite images are named SFB and NR respectively as shown in Fig. 3c and d. The image SFB is acquired by the Thematic Mapper on Landsat 5 on September 9, 2002 while image NR is an astronaut photograph taken by Kodak 760 C digital camera on November 8, 2006. Size of the image SFB and NR are  $512 \times 512$  and  $539 \times 539$ , respectively. Due to the two satellite images are multi-spectral images, we transformed the two images using a famous orthogonal transformation method, i.e., principal component analysis (PCA). As the most informative component, the first principal component is selected for multilevel image thresholding in this chapter.

## 5.2 Experimental Settings

The relative parameter settings of algorithms utilized in this chapter are shown in Table 1. As illustrated in Table 1, to perform a fair experiment, in all of the seven algorithms, the population size ( $N$ ) and maximum number of fitness evaluations ( $FES$ ) were set to 30 and 3000, respectively. For GSA-GA, GSA, PSO-GSA, and GGSA, the initial gravitational constant  $G_0$  and decrease coefficient  $\beta$  were set to 100 and 20 as the default values in original GSA [56]. For PSO-GSA, the two acceleration coefficients ( $c_1$  and  $c_2$ ) were set to 0.5 and 1.5, respectively as suggested in [66]. For GGSA, the two acceleration coefficients ( $c_1$  and  $c_2$ ) were set to  $-2t^3/Iter_{\max}^3 + 2$  and

**Table 1** Parameter settings of the compared algorithms in this chapter

Parameters	GSA-GA	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO
$FES$	300	300	300	300	300	300	300
$N$	30	30	30	30	30	30	30
$p_m$	0.1	–	–	–	–	DE/rand/1	0.1
$p_c/CR$	–	–	–	–	–	0.9	1
$F$	–	–	–	–	–	0.5	–
$G_0$	100	100	100	100	–	–	–
$\beta$	20	20	20	20	–	–	–
$c_1$	–	–	0.5	$-2t^3/$ $Iter_{max}^3 + 2$	–	–	2
$c_2$	–	–	1.5	$2t^3/Iter_{max}^3$	–	–	2
$p_1$	–	–	–	–	0.3*rand	–	–
$p_2$	–	–	–	–	0.3*rand	–	–
$n_{elit}$	–	–	–	–	–	2	–

$2t^3/Iter_{max}^3$ , respectively as recommended in [67]. The other parameters in DS, BBO-DE, and GAPSO were all adopted as the suggested values in the original papers. To be specific, in DS, the two control parameters (i.e.,  $p_1$  and  $p_2$ ) were both set to 0.3\*rand; in BBO-DE, the mutation scheme was the DE/rand/1, the crossover probability ( $CR$ ), the scaling factor ( $F$ ), and the elitism parameter ( $n_{elit}$ ) were set to 0.9, 0.5, and 2 [40]; in GAPSO, the mutation probability ( $p_m$ ) was set to 0.1, the crossover probability (denoted by  $p_c$ ) was set to 1, and the two acceleration coefficients were both set to 2 [36]. For GSA-GA, the mutation probability  $p_m$  was set to 0.1 as it is recommended in many GA based algorithms [23, 37, 40].

### 5.3 Performance Metrics

In this chapter, the uniformity measure [69] is adopted to qualitatively judge the segmentation results based on different thresholds. The uniformity measure ( $u$ ) is calculated as follows:

$$u = 1 - 2 \times m \times \frac{\sum_{j=1}^{m+1} \sum_{i \in Re_j} (f_i - g_j)^2}{Num \times (f_{max} - f_{min})^2}, \quad (10)$$

where  $m$  is the number of desired thresholds,  $Re_j$  is the  $j$ -th segmented region,  $Num$  is the total number of pixels in the given image  $IMG$ ,  $f_i$  is the gray-level of pixel  $i$ ,  $g_j$  is the mean gray-level of pixels in  $j$ -th region,  $f_{max}$  and  $f_{min}$  are the maximum and minimum gray-level of pixels in the given image, respectively. Typically,

$u \in [0, 1]$  and a higher value of uniformity indicates that there is better uniformity in the thresholding image.

#### 5.4 Experiment 1: Maximizing Kapur's Entropy

The Kapur's entropy criterion was first maximized to construct an optimally segmented image for all the seven NAs. Table 2 presented the best uniformity ( $u$ ) and the corresponding threshold values ( $Th$ ) on all the test images produced by the algorithms after 30 independent runs. Moreover, to test the stability of thresholding methods, average uniformity of 30 independent runs are reported in Table 3. Besides, to test the computation complexity, the mean of CPU times are reported in Table 4.

As shown in Table 2, for image House, GSA-GA outperformed GSA, PSOGSA, GGSA, DS, and GAPSO on all the different number of thresholds. Although BBO-DE gained the best uniformity when  $m = 4$ , GSA-GA has a better mean uniformity as illustrated in Table 3. For image Pepper, SFB and NR, GSA-GA performed as well as or was better than the other GSA-based segmentation methods on the best uniformity reported in Table 2. Meanwhile, mean uniformity of the three images shown in Table 3 also confirmed the superiority of GSA-GA in most cases. This may be due to that the introduced mutation operator prevents GSA from getting stuck to local optima. Furthermore, when comparing with DS, BBO-DE, and GAPSO, GSA-GA displayed the best on both the best and mean uniformity as presented in Tables 2 and 3. That is to say, GSA variant-based multilevel image thresholding methods are more efficient than other NAs based techniques. That is because particles in GSA variants can learn from other particles fuller than other NAs. Besides, it is worth noting that the mean CPU times of GSA-GA are visible smaller than the comparison algorithms as reported in Table 4.

For a visual interpretation of the segmentation results, the segmented images of the two satellite images obtained based on the optimum thresholds shown in Table 2 were presented in Figs. 4 and 5 with  $m = 2 - 5$ . For the two images, it is hard to say which number of thresholds is more suitable for due to the complexity of these images. Their inherent uncertainty and ambiguity make it a challenging task for these tested algorithms to determine the proper number of classes in a given image.

#### 5.5 Experiment 2: Maximizing Otsu

This section devotes to maximize the Otsu criterion to construct an optimally segmented image for all of the seven algorithms. Experiments were performed on all the test images shown in Fig. 3. The best uniformity ( $u$ ) and the corresponding threshold values ( $Th$ ) produced by the algorithms were given in Table 5 after 30 independent runs. Similar to Sect. 5.4, to test the stability of the algorithms, Table 6 presented the mean of uniformity. In addition, Table 7 provided the mean of CPU times.

**Table 2** Comparison of algorithms taking the Kapur's entropy as evaluation criterion in terms of best uniformity and corresponding thresholds

Image	$m$	GSA		PSOGSA		GGSA		
		$Th$	$u$	$Th$	$u$	$Th$	$u$	
House	2	103, 197	0.9302	98, 194	0.9265	98, 193	0.9268	
	3	64, 116, 206	0.9146	65, 112, 194	0.9109	65, 112, 193	0.9113	
	4	59, 97, 132, 193	0.9728	59, 88, 115, 194	0.8932	61, 101, 137, 183	0.9844	
	5	66, 101, 134, 164, 201	0.9886	61, 101, 137, 183, 195	0.9868	61, 101, 137, 182, 194	0.9869	
	2	71, 138	<b>0.9783</b>	71, 138	<b>0.9783</b>	71, 138	<b>0.9783</b>	
Pepper	3	64, 121, 169	<b>0.9817</b>	65, 122, 169	<b>0.9817</b>	64, 121, 168	<b>0.9817</b>	
	4	49, 89, 129, 171	<b>0.9843</b>	71, 138, 152, 253	0.9656	49, 88, 129, 172	<b>0.9843</b>	
	5	52, 79, 116, 151, 183	0.9856	65, 122, 152, 169, 252	0.9742	62, 111, 141, 174, 219	0.9777	
	2	92, 156	<b>0.9719</b>	92, 156	<b>0.9719</b>	92, 156	<b>0.9719</b>	
	3	83, 138, 187	<b>0.9760</b>	83, 139, 191	<b>0.9760</b>	82, 140, 189	0.9759	
SFB	4	76, 125, 163, 202	0.9785	92, 102, 156, 195	0.9687	72, 114, 152, 200	<b>0.9788</b>	
	5	60, 108, 136, 171, 208	0.9817	38, 83, 102, 139, 191	0.9723	89, 94, 138, 162, 197	0.9714	
	2	89, 137	<b>0.9865</b>	89, 137	<b>0.9865</b>	89, 138	0.9864	
	3	78, 105, 145	0.9885	79, 106, 146	<b>0.9886</b>	79, 106, 146	<b>0.9886</b>	
	4	73, 97, 118, 155	<b>0.9896</b>	89, 131, 137, 254	0.9752	89, 126, 137, 253	0.9772	
NR	5	67, 87, 104, 124, 158	<b>0.9902</b>	79, 106, 131, 146, 253	0.9841	74, 103, 131, 140, 253	0.9824	
	DS		BBO-DE		GAPSO		GSA-GA	
	$Th$	$u$	$Th$	$u$	$Th$	$u$	$Th$	$u$
	112, 196	0.9351	68, 99	0.9170	111, 194	0.9354	67, 123	<b>0.9446</b>
	63, 110, 200	0.9077	104, 184, 196	0.9785	64, 133, 195	0.9630	109, 152, 196	<b>0.9830</b>
63, 104, 137, 185	0.9828	51, 96, 163, 205	<b>0.9834</b>	52, 82, 145, 197	0.9786	51, 106, 161, 198	<b>0.9834</b>	
36, 78, 107, 149, 201	0.9844	46, 98, 127, 179, 199	0.9839	66, 90, 126, 180, 192	0.9839	70, 103, 139, 177, 202	<b>0.9887</b>	

(continued)

**Table 2** (continued)

DS	BBO-DE		GAPSO		GSA-GA	
	<i>u</i>	<i>Th</i>	<i>u</i>	<i>Th</i>	<i>u</i>	<i>Th</i>
75, 142	0.9781	76, 146	0.9776	68, 136	0.9782	71, 138
60, 133, 176	0.9800	57, 122, 173	0.9810	66, 114, 172	0.9803	64, 123, 170
49, 111, 134, 182	0.9789	45, 99, 121, 152	0.9771	53, 115, 153, 185	0.9806	38, 83, 128, 171
57, 98, 126, 177, 222	0.9781	44, 83, 131, 161, 172	0.9827	55, 115, 148, 171, 249	0.9758	42, 81, 117, 153, 183
88, 153	0.9718	84, 155	0.9715	93, 170	0.9697	93, 157
76, 126, 185	0.9740	79, 151, 187	0.9724	100, 144, 190	0.9731	83, 138, 189
93, 141, 181, 206	0.9742	64, 110, 160, 201	0.9770	56, 98, 145, 226	0.9705	73, 115, 154, 202
60, 98, 120, 164, 194	0.9767	45, 97, 132, 175, 212	0.9786	85, 108, 133, 174, 237	0.9757	68, 107, 140, 176, 218
88, 139	0.9864	82, 135	0.9857	85, 138	0.9861	89, 137
81, 108, 136	0.9875	75, 97, 142	0.9873	93, 118, 166	0.9846	79, 106, 146
85, 118, 154, 191	0.9847	67, 99, 117, 160	0.9877	75, 97, 145, 204	0.9850	80, 108, 127, 157
77, 102, 121, 156, 191	0.9890	80, 84, 103, 143, 183	0.9854	76, 98, 141, 150, 215	0.9832	77, 89, 109, 123, 159

*u*

*Th*

*u*

*Th*

*u*

*Th*

*u*

*Th*

*u*

**Table 3** Comparison of algorithms taking the Kapur's entropy as evaluation criterion in terms of the mean of uniformity

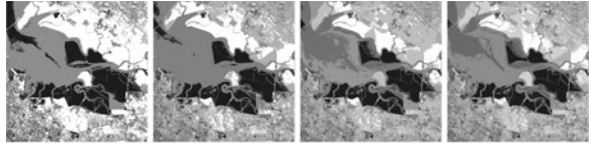
Image	$m$	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO	GSA-GA
House	2	0.9185	0.9265	0.9267	0.9109	0.9103	0.9162	<b>0.9300</b>
	3	0.9084	0.9109	0.9108	0.8882	0.9366	0.9084	<b>0.9370</b>
	4	0.9064	0.8869	0.9129	0.9306	0.9218	0.9288	<b>0.9311</b>
	5	0.9870	0.9366	0.9846	0.9322	0.9191	0.9805	<b>0.9877</b>
Pepper	2	<b>0.9783</b>	<b>0.9783</b>	<b>0.9783</b>	0.9764	0.9757	0.9778	<b>0.9783</b>
	3	0.9816	0.9700	<b>0.9817</b>	0.9755	0.9779	0.9782	0.9816
	4	0.9793	0.9643	0.9707	0.9748	0.9740	0.9769	<b>0.9809</b>
	5	0.9769	0.9669	0.9694	0.9768	0.9786	0.9718	<b>0.9803</b>
SFB	2	<b>0.9719</b>	<b>0.9719</b>	0.9718	0.9692	0.9700	0.9682	<b>0.9719</b>
	3	0.9757	0.9652	0.9751	0.9688	0.9709	0.9703	<b>0.9760</b>
	4	0.9773	0.9556	0.9704	0.9716	0.9737	0.9689	<b>0.9785</b>
	5	0.9740	0.9648	0.9658	0.9717	0.9732	0.9731	<b>0.9766</b>
NR	2	<b>0.9865</b>	<b>0.9865</b>	0.9864	0.9859	0.9845	0.9845	<b>0.9865</b>
	3	0.9885	0.9749	0.9877	0.9838	0.9818	0.9835	<b>0.9886</b>
	4	<b>0.9891</b>	0.9752	0.9732	0.9824	0.9827	0.9823	0.9869
	5	0.9860	0.9833	0.9771	0.9857	0.9817	0.9808	<b>0.9862</b>

**Table 4** Comparison of algorithms taking the Kapur's entropy as evaluation criterion in terms of the mean of CPU times (in seconds)

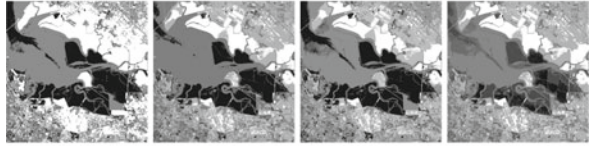
Image	$m$	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO	GSA-GA
House	2	1.0488	1.0765	1.0323	1.1322	2.3825	1.0707	<b>0.9144</b>
	3	1.1113	1.0908	1.1569	1.1852	2.5713	1.1439	<b>0.9246</b>
	4	1.0932	1.1090	1.1449	2.0981	2.8502	1.1865	<b>0.9645</b>
	5	1.2845	1.1854	1.1890	1.9882	2.9229	1.2625	<b>1.0268</b>
Pepper	2	0.7239	0.7881	0.7153	0.6398	1.5238	1.7950	<b>0.4939</b>
	3	0.7545	0.7495	0.7318	0.6501	1.5997	0.7175	<b>0.5024</b>
	4	0.7661	0.7739	0.7559	0.6829	1.6234	0.6883	<b>0.5288</b>
	5	0.7680	0.7942	0.7950	0.6981	1.7266	0.7097	<b>0.5375</b>
SFB	2	1.4644	1.4473	1.4620	1.4138	2.9996	1.4216	<b>1.0782</b>
	3	1.4855	1.4803	1.4838	1.4334	3.0790	1.5437	<b>1.0633</b>
	4	1.4879	1.4911	1.5245	1.4626	3.7202	1.4634	<b>1.0861</b>
	5	1.5132	1.5120	1.5095	1.4603	3.2803	1.6632	<b>1.0911</b>
NR	2	1.5570	1.5690	1.5323	1.5154	3.2308	1.5084	<b>1.1575</b>
	3	1.5604	1.5729	1.5594	1.5209	3.2722	1.5922	<b>1.2417</b>
	4	1.5937	1.5997	1.5696	1.5484	3.3852	1.5545	<b>1.1933</b>
	5	1.5952	1.5852	1.6206	1.5784	3.4412	1.5969	<b>1.2306</b>



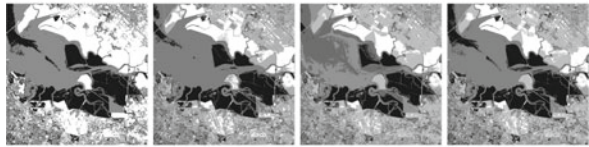
**Fig. 4** Segmented images for the SFB image by Kapur's entropy with different optimization techniques



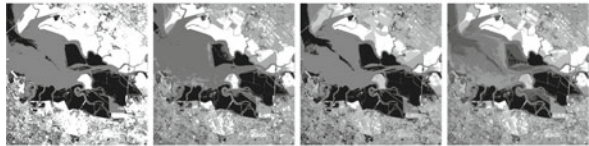
(a) GSA based thresholding results with  $m=2, 3, 4$  and  $5$



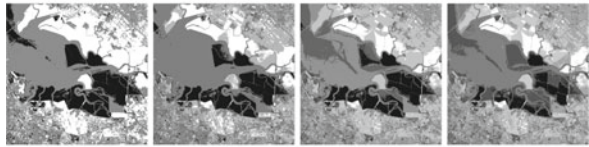
(b) PSO based thresholding results with  $m=2, 3, 4$  and  $5$



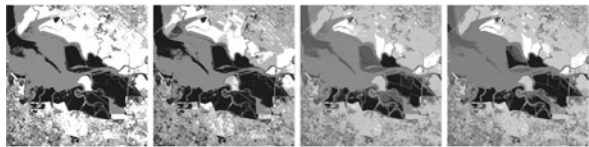
(c) GGSA based thresholding results with  $m=2, 3, 4$  and  $5$



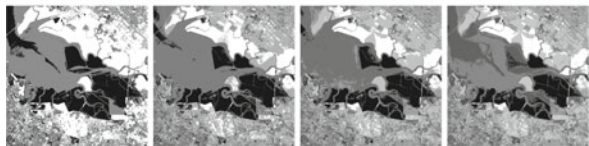
(d) DS based thresholding results with  $m=2, 3, 4$  and  $5$



(e) BBODE based thresholding results with  $m=2, 3, 4$  and  $5$

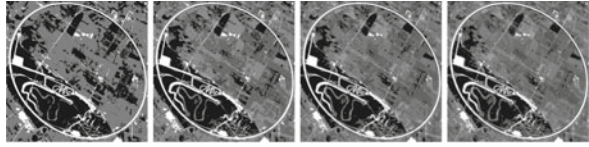


(f) GAPSO based thresholding results with  $m=2, 3, 4$  and  $5$

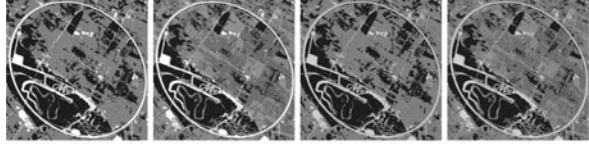


(g) GSA-GA based segmented images with  $m=2, 3, 4$  and  $5$

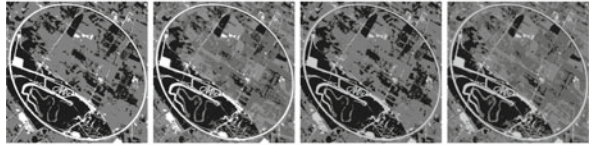
**Fig. 5** Segmented images for the NR image by Kapur's entropy with different optimization techniques



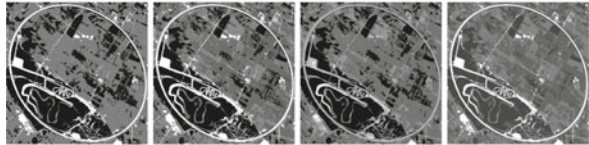
(a) GSA based thresholding results with  $m=2, 3, 4$  and  $5$



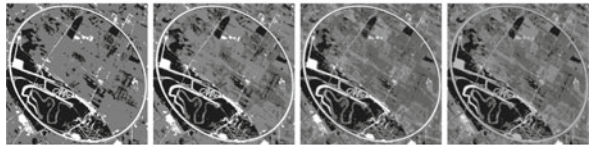
(b) PSO based thresholding results with  $m=2, 3, 4$  and  $5$



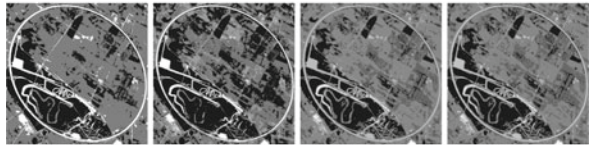
(c) GGSA based thresholding results with  $m=2, 3, 4$  and  $5$



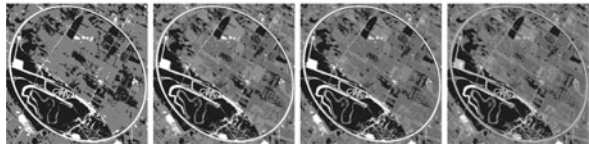
(d) DS based thresholding results with  $m=2, 3, 4$  and  $5$



(e) BBODE based thresholding results with  $m=2, 3, 4$  and  $5$



(f) GAPSO based thresholding results with  $m=2, 3, 4$  and  $5$



(g) GSA-GA based thresholding results with  $m=2, 3, 4$  and  $5$

**Table 5** Comparison of algorithms taking the Otsu as evaluation criterion in terms of best uniformity and corresponding thresholds

Image	$m$	GSA		PSOGSA		GGSA	
		$Th$	$u$	$Th$	$u$	$Th$	$u$
House	2	96, 155	<b>0.9875</b>	96, 155	<b>0.9875</b>	96, 155	<b>0.9875</b>
	3	79, 110, 154	0.9862	90, 109, 147	0.9847	81, 112, 158	<b>0.9864</b>
	4	81, 109, 153, 201	0.9875	96, 109, 155, 197	0.9851	78, 93, 109, 149	0.9823
	5	80, 102, 115, 154, 194	0.9812	96, 108, 152, 199, 204	0.9815	80, 110, 148, 164, 238	0.9814
	2	77, 147	0.9774	77, 148	0.9772	77, 148	0.9772
Pepper	3	63, 113, 164	0.9809	63, 113, 163	0.9809	63, 113, 163	0.9809
	4	50, 88, 131, 173	<b>0.9843</b>	39, 87, 132, 174	0.9834	49, 87, 128, 172	<b>0.9843</b>
	5	43, 78, 118, 162, 197	0.9828	45, 82, 113, 145, 180	<b>0.9860</b>	48, 85, 118, 157, 197	0.9831
	2	87, 175	0.9662	86, 176	0.9653	86, 176	0.9653
	3	82, 145, 201	0.9754	80, 140, 199	0.9755	80, 140, 199	0.9755
SFB	4	66, 109, 155, 202	<b>0.9782</b>	68, 109, 155, 200	<b>0.9782</b>	57, 99, 149, 199	0.9765
	5	50, 86, 128, 165, 206	0.9794	52, 89, 128, 165, 204	0.9799	51, 87, 127, 164, 204	0.9795
	2	131, 193	0.9684	132, 193	0.9681	132, 193	0.9681
	3	86, 133, 193	0.9821	84, 133, 193	0.9818	84, 133, 193	0.9818
	4	88, 136, 181, 221	0.9774	82, 122, 156, 196	0.9830	84, 132, 178, 219	0.9779
NR	5	71, 106, 148, 192, 220	0.9822	77, 106, 133, 163, 196	<b>0.9873</b>	80, 116, 147, 191, 223	0.9815
	DS		GAPSO		GSA-GA		
	$Th$	$Th$	$u$	$Th$	$u$	$Th$	$u$
	103, 157	91, 145	0.9867	99, 156	0.9874	96, 155	<b>0.9875</b>
	81, 109, 156	81, 106, 165	0.9856	74, 110, 169	0.9853	80, 111, 158	<b>0.9864</b>
76, 106, 162, 204	85, 109, 133, 164	0.9855	83, 125, 160, 216	0.9850	83, 113, 155, 202	<b>0.9876</b>	

(continued)

Table 5 (continued)

DS	BBO-DE		GAPSO		GSA-GA	
	<i>u</i>	<i>Th</i>	<i>u</i>	<i>Th</i>	<i>u</i>	<i>Th</i>
78, 91, 127, 175, 213	0.9848	66, 104, 137, 177, 199	0.9883	90, 115, 152, 184, 212	0.9870	76, 102, 132, 167, 199
65, 136	<b>0.9781</b>	72, 122	0.9754	59, 127	0.9769	77, 148
65, 134, 166	0.9797	59, 113, 183	0.9766	53, 109, 156	0.9794	61, 121, 166
72, 109, 145, 181	0.9804	50, 71, 136, 169	0.9791	67, 107, 145, 184	0.9811	51, 88, 130, 173
42, 62, 94, 128, 175	0.9821	77, 113, 141, 160, 185	0.9770	56, 99, 143, 186, 192	0.9772	40, 83, 117, 161, 197
79, 150	<b>0.9710</b>	92, 174	0.9678	83, 162	0.9705	83, 161
89, 139, 193	0.9756	73, 134, 193	0.9749	90, 154, 199	0.9735	81, 138, 195
76, 110, 165, 204	0.9750	79, 103, 132, 204	0.9707	60, 110, 166, 218	0.9734	64, 107, 151, 200
37, 95, 144, 160, 198	0.9723	58, 104, 119, 154, 208	0.9770	69, 102, 127, 190, 205	0.9703	72, 102, 136, 177, 213
126, 199	0.9692	120, 171	0.9737	90, 136	<b>0.9864</b>	127, 194
92, 131, 200	0.9815	85, 126, 175	<b>0.9843</b>	84, 145, 185	0.9802	84, 131, 193
91, 125, 153, 203	0.9817	81, 117, 141, 177	<b>0.9850</b>	89, 117, 147, 201	0.9841	81, 117, 141, 177
89, 99, 129, 156, 194	0.9832	80, 107, 155, 180, 230	0.9827	86, 121, 162, 189, 224	0.9792	77, 106, 133, 163, 196

**Table 6** Comparison of algorithms taking the Otsu as evaluation criterion in terms of the mean of uniformity

Image	<i>m</i>	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO	GSA-GA
House	2	<b>0.9875</b>	<b>0.9875</b>	<b>0.9875</b>	0.9830	0.9864	0.9867	<b>0.9875</b>
	3	0.9848	0.9758	0.9837	0.9849	0.9843	0.9847	<b>0.9863</b>
	4	0.9811	0.9695	0.9791	0.9840	0.9837	0.9839	<b>0.9866</b>
	5	0.9812	0.9757	0.9802	0.9834	<b>0.9859</b>	0.9833	0.9853
Pepper	2	0.9768	0.9768	<b>0.9772</b>	0.9714	0.9722	0.9728	<b>0.9772</b>
	3	0.9808	<b>0.9809</b>	<b>0.9809</b>	0.9723	0.9723	0.9775	<b>0.9809</b>
	4	0.9803	0.9830	0.9838	0.9752	0.9750	0.9769	<b>0.9839</b>
	5	0.9820	<b>0.9836</b>	0.9824	0.9750	0.9699	0.9753	0.9814
SFB	2	0.9657	0.9653	0.9653	0.9634	0.9628	0.9683	<b>0.9664</b>
	3	0.9753	<b>0.9755</b>	0.9754	0.9701	0.9717	0.9580	0.9746
	4	0.9760	0.9726	0.9761	0.9675	0.9656	0.9594	<b>0.9763</b>
	5	0.9765	0.9768	0.9765	0.9621	0.9738	0.9661	<b>0.9792</b>
NR	2	<b>0.9682</b>	0.9681	0.9681	0.9657	0.9685	0.9756	0.9679
	3	<b>0.9818</b>	0.9813	0.9817	0.9763	0.9755	0.9748	<b>0.9818</b>
	4	0.9752	0.9770	0.9762	0.9754	0.9763	0.9733	<b>0.9775</b>
	5	0.9768	0.9803	0.9802	0.9666	0.9710	0.9745	<b>0.9809</b>

**Table 7** Comparison of algorithms taking the Otsu as evaluation criterion in terms of the mean of CPU times (in seconds)

Image	<i>m</i>	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO	GSA-GA
House	2	1.1504	0.9398	0.8722	1.0113	2.2981	0.9848	<b>0.7026</b>
	3	0.9435	0.9785	0.8744	0.9876	2.3724	1.0608	<b>0.7657</b>
	4	1.0936	1.0273	1.1108	1.1089	2.5155	1.0065	<b>0.7762</b>
	5	1.0838	1.0335	1.0464	1.2391	2.6006	1.0453	<b>0.7325</b>
Pepper	2	0.7789	0.7764	0.7861	0.7115	1.5719	0.6825	<b>0.6266</b>
	3	0.8076	0.8006	0.8307	0.7262	1.8429	0.7527	<b>0.6437</b>
	4	0.9038	0.8315	0.9399	0.7397	2.0726	0.7486	<b>0.6428</b>
	5	0.8507	0.8810	0.8743	0.7648	1.8911	0.7573	<b>0.6644</b>
SFB	2	1.5234	1.4966	1.4985	1.5176	3.5485	1.4641	<b>1.3279</b>
	3	1.5384	1.5032	1.5652	1.5036	3.2132	1.4819	<b>1.2740</b>
	4	1.5683	1.5484	1.6148	1.6120	3.6203	1.5370	<b>1.3008</b>
	5	1.5844	1.5753	1.6248	1.5562	3.3839	1.4967	<b>1.1065</b>
NR	2	1.5666	1.5938	1.6228	1.5742	3.2607	1.6392	<b>1.4363</b>
	3	1.5838	1.6257	1.6352	1.6902	4.0370	1.6036	<b>1.4148</b>
	4	1.6289	1.7799	1.6631	1.6329	3.4892	1.7265	<b>1.4367</b>
	5	1.6597	1.6547	1.7577	1.6487	3.6371	1.6850	<b>1.4455</b>

Generally speaking, with the Otsu criterion, GSA-GA yielded better results than the other six comparison algorithms as illustrated in Table 5. Moreover, similar to the results given in Table 4, the mean uniformity reported in Table 6 also confirms the superiority of GSA-based methods comparing with other NAs. Mean CPU times illustrated in Table 7 showed that the computational time of GSA-GA is also the lowest. Especially, time consuming of GSA-GA is much lower than that of DS. In Figs. 6 and 7, the segmentation images of SFB and NR by all the seven NAs used the Otsu criterion are presented.

Furthermore, comparing the mean uniformity shown in Tables 3 and 6, we can concluded that for a given image, thresholds determined based on different criteria can be different. For images Pepper, SFB, and NR, the thresholding results using Otsu criterion is better than the results on the basis of Kapur's entropy. While for the image House, we observed completely opposite conclusions. Thereby, comprehensive analysis for choosing the most appropriate objective function are desired in the future real-world application.

## 5.6 Running Time Analysis Using Student's *t*-test

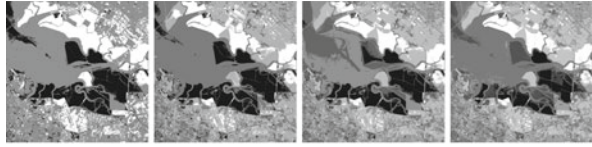
To statistically analyze the time-consuming shown in Tables 4 and 7, a parametric significance proof known as the Student's *t*-test was conducted in this section [55]. This test allows assessing result differences between two related methods (one of the methods is chosen as control method). In this chapter, the control algorithm is GSA-GA. Results of GSA-GA were compared with other six comparison NAs in terms of the mean CPU times. The null hypothesis in this chapter is that there is no significance difference between the mean CPU times achieved by two compared algorithms for a test image over 30 independent runs. The significance level is  $\alpha = 0.05$ . Accordingly, if the produced *t*-value of a test is smaller than or equal to the critical value “-2.045”, the null hypothesis for the paired *t*-test should be rejected [70, 71].

Tables 8 and 9 reported the *t*-values produced by Student's *t*-test for the pairwise comparison of six groups. These groups were formed by GSA-GA (with Kapur's entropy and Otsu, respectively) versus other comparison algorithms. In Tables 8 and 9, if the null hypothesis is rejected, we define a symbol “ $h = 0$ ” which means the difference between the results obtained by two algorithms are not different. By contrast, “ $h = 1$ ” indicates that the difference between GSA-GA and the compared algorithm is significant.

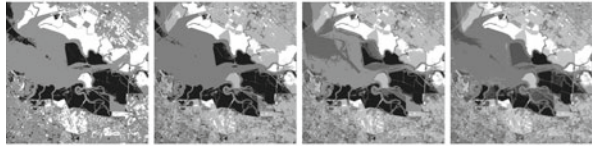
As illustrated in Tables 8 and 9, for all the test images, the produced *t*-values for the experiments on the Kapur's entropy and Otsu criteria were smaller than the critical value. The results indicated that GSA-GA performed significantly fast image segmentation compared with the other six algorithms.



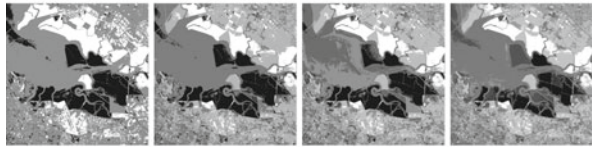
**Fig. 6** Segmented images for the SFB image by Otsu with different optimization techniques



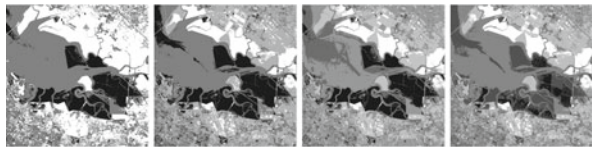
(a) GSA based thresholding results with  $m=2, 3, 4$  and  $5$



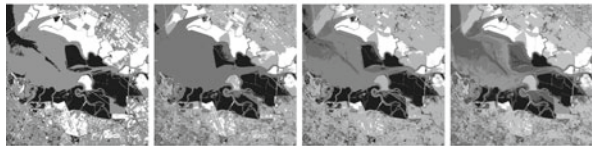
(b) PSO based thresholding results with  $m=2, 3, 4$  and  $5$



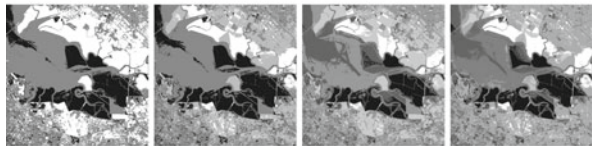
(c) GGSA based thresholding results with  $m=2, 3, 4$  and  $5$



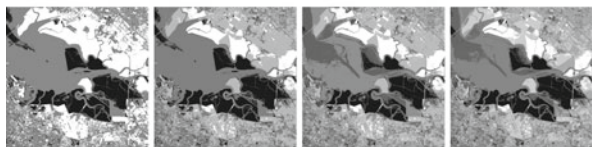
(d) DS based thresholding results with  $m=2, 3, 4$  and  $5$



(e) BBO-DE based thresholding results with  $m=2, 3, 4$  and  $5$

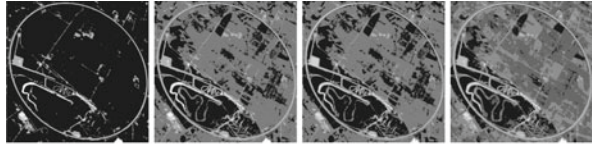


(f) GAPSO based thresholding results with  $m=2, 3, 4$  and  $5$

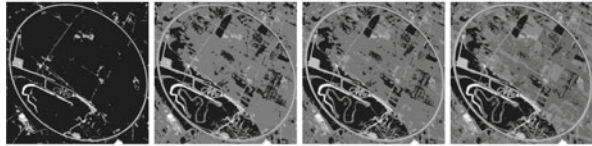


(g) GSA-GA based segmented images with  $m=2, 3, 4$  and  $5$

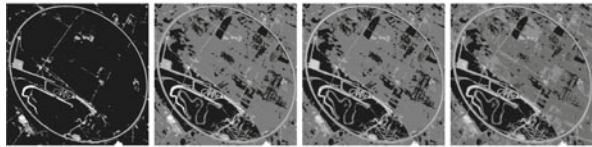
**Fig. 7** Segmented images for the NR image by Otsu with different optimization techniques



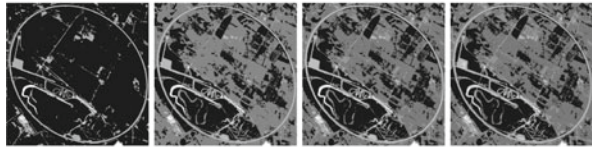
(a) GSA based thresholding results with  $m=2, 3, 4$  and  $5$



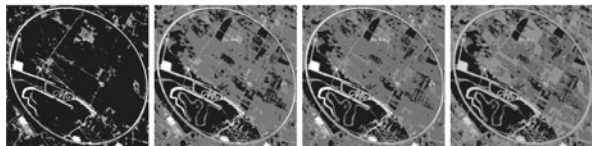
(b) PSOGSA based thresholding results with  $m=2, 3, 4$  and  $5$



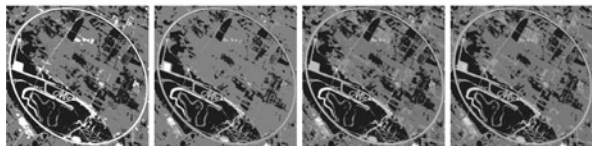
(c) GGSA based thresholding results with  $m=2, 3, 4$  and  $5$



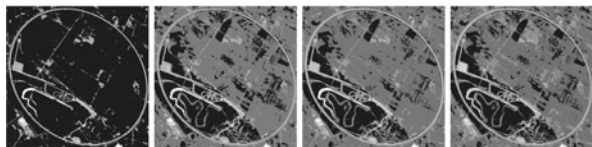
(d) DS based thresholding results with  $m=2, 3, 4$  and  $5$



(e) BBO-DE based thresholding results with  $m=2, 3, 4$  and  $5$



(f) GAPSO based thresholding results with  $m=2, 3, 4$  and  $5$



(g) GSA-GA based segmented images with  $m=2, 3, 4$  and  $5$



**Table 8** Statistical analysis of 30 runs for the experimental results on the Kapur's entropy criterion

Image	<i>m</i>	GSA-GA VS					DS	BBO-DE	GAPSO				
		GSA	PSOGSA	GGSA	DS	BBO-DE							
House	2	-15.1156	1	-19.1332	1	-14.2639	1	-24.7401	1	-107.8601	1	-24.4811	1
	3	-32.6476	1	-19.9512	1	-26.7523	1	-44.5436	1	-109.2128	1	-29.0906	1
	4	-39.5957	1	-24.2611	1	-20.0202	1	-44.2350	1	-102.7793	1	-46.5004	1
	5	-34.7283	1	-30.5527	1	-43.5916	1	-26.5975	1	-222.9597	1	-39.2664	1
	Pepper	2	-10.7081	1	-10.4253	1	-11.1125	1	-5.9857	1	-65.2916	1	-3.8069
SFB	3	-64.8463	1	-57.2600	1	-55.2033	1	-29.4831	1	-94.3847	1	-30.4911	1
	4	-36.0189	1	-42.7815	1	-26.6089	1	-23.4535	1	-33.6639	1	-24.6679	1
	5	-2.5627	1	-2.9435	1	-2.8870	1	-1.3815	1	-16.6481	1	-1.2784	1
	2	-27.6977	1	-24.3816	1	-29.6109	1	-12.1633	1	-45.1326	1	-23.9737	1
	3	-22.4245	1	-21.4709	1	-23.8949	1	-21.7750	1	-151.1436	1	-19.7951	1
NR	4	-14.6253	1	-13.5423	1	-17.3410	1	-11.0828	1	-79.4278	1	-10.7647	1
	5	-75.6004	1	-57.9269	1	-54.1767	1	-66.5839	1	-341.5912	1	-50.8013	1
	2	-21.8825	1	-22.3767	1	-23.0353	1	-22.0319	1	-278.2518	1	-12.4801	1
	3	-10.3605	1	-12.4493	1	-13.3727	1	-11.0882	1	-71.6375	1	-11.6034	1
	4	-13.2716	1	-11.2340	1	-14.4755	1	-13.4430	1	-144.0898	1	-14.9473	1
5	-12.9729	1	-12.7915	1	-16.0183	1	-11.9100	1	-111.5124	1	-13.4166	1	

**Table 9** Statistical analysis of 30 runs for the experimental results on the Otsu criterion

Image	<i>m</i>	GSA-GA VS											
		GSA		PSOGSA		GGSA		DS		BBO-DE		GAPSO	
		<i>t</i> -value	<i>h</i>	<i>t</i> -value	<i>h</i>	<i>t</i> -value	<i>h</i>	<i>t</i> -value	<i>h</i>	<i>t</i> -value	<i>h</i>	<i>t</i> -value	<i>h</i>
House	2	-9.5851	1	-11.7822	1	-9.4314	1	-14.6542	1	-64.9566	1	-11.9011	1
	3	-20.6436	1	-15.8776	1	-18.2415	1	-20.7056	1	-133.0959	1	-22.5299	1
	4	-14.1373	1	-19.2192	1	-27.5020	1	-11.2347	1	-45.3072	1	-28.2547	1
	5	-35.3391	1	-22.4042	1	-27.7047	1	-49.8531	1	-120.2696	1	-38.4260	1
	2	-65.6607	1	-36.0491	1	-70.6444	1	-41.2547	1	-58.9202	1	-3.5054	1
Pepper	3	-80.3597	1	-73.7771	1	-84.5025	1	-56.4896	1	-97.0729	1	-53.7705	1
	4	-51.9612	1	-44.2863	1	-44.7699	1	-21.8072	1	-205.1454	1	-18.4475	1
	5	-50.4190	1	-56.5007	1	-52.2792	1	-35.4027	1	-92.6841	1	-18.5975	1
	2	-40.5912	1	-40.1226	1	-40.2393	1	-35.7284	1	-213.5478	1	-39.6544	1
	3	-40.0638	1	-47.2556	1	-45.0096	1	-39.0400	1	-202.4002	1	-21.6851	1
SFB	4	-37.4041	1	-36.5145	1	-41.7963	1	-36.6461	1	-92.7530	1	-35.9711	1
	5	-55.9229	1	-51.7381	1	-53.8157	1	-55.1532	1	-170.0324	1	-36.5197	1
	2	-81.1067	1	-70.7878	1	-73.9369	1	-57.9793	1	-21.2059	1	-77.3845	1
	3	-51.7581	1	-78.3780	1	-66.3253	1	-77.0030	1	-390.4326	1	-27.7027	1
	4	-41.9257	1	-48.8776	1	-45.0429	1	-40.2674	1	-25.7615	1	-44.0133	1
5	-43.5523	1	-43.6292	1	-46.6667	1	-42.4596	1	-231.6469	1	-35.1170	1	

## 6 Conclusion

Multilevel thresholding is a fundamental and important technology for image segmentation and a fundamental process in image interpretation. An overview of the grayscale image segmentation using multilevel thresholding and nature-inspired algorithms is discussed in this chapter. On one hand, the NAs based multilevel image thresholding can reduce the computational consuming of exhaustive search algorithms based techniques. On the other hand, the global optimization property of NAs makes them as preferable choices for multilevel thresholding. Although some drawbacks of the NAs limited their application, hybrid algorithms can promote their performances. Therefore, a novel multilevel thresholding algorithm for image segmentation by employing an improved GSA variant, called GSA-GA is proposed in this chapter. In GSA-GA, the roulette selection and discrete mutation operators of GA were introduced into GSA to improve the search accuracy and speed of GSA. Experiments on benchmark images and real-world satellite images confirmed the superiority of GSA-GA compared with GSA, PSO, GSA, GGSA, DS, BBO-DE, and GAPSO. Potential future investigation can be on the analysis of different criteria. By comprehensive analysis, the most appropriate criterion for constructing objective function can be selected or developed.

**Acknowledgments** This work was supported by the Chinese Natural Science Foundation Projects (No. 41471353, No. 41271349), Fundamental Research Funds for the Central Universities (No. 14CX02039A, No. 15CX06001A), and the China Scholarship Council.

## References

1. Zhang, A.Z., Sun, G.Y., Wang, Z.J., Yao, Y.J.: A hybrid genetic algorithm and gravitational search algorithm for global optimization. *Neural. Netw. World.* **25**, 53–73 (2015)
2. Ghamisi, P., Couceiro, M.S., Martins, F.M., Atli, B.J.: Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization. *IEEE. Trans. Geosci. Remote. Sens.* **52**, 2382–2394 (2014)
3. Kang, W.X., Yang, Q.Q., Liang, R.P.: The comparative research on image segmentation algorithms. In: 2009 First International Workshop on Education Technology and Computer Science, pp. 703–707. IEEE (2009)
4. Bhandari, A.K., Singh, V.K., Kumar, A., Singh, G.K.: Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert. Syst. Appl.* **41**, 3538–3560 (2014)
5. Dey, S., Bhattacharyya, S., Maulik, U.: Quantum Behaved Multi-objective PSO and ACO Optimization for Multi-level Thresholding. IEEE, New York (2014)
6. Hornig, M.H.: Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. *Expert. Syst. Appl.* **38**, 13785–13791 (2011)
7. Bhandari, A.K., Kumar, A., Singh, G.K.: Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. *Expert. Syst. Appl.* **42**, 8707–8730 (2015)
8. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE. Trans. Syst. Man Cybern.* **9**, 62–66 (1979)

9. Kapur, J.N., Sahoo, P.K., Wong, A.K.: A new method for gray-level picture thresholding using the entropy of the histogram. *Comput. Vision Graph.* **29**, 273–285 (1985)
10. Huang, L.K., Wang, M.J.J.: Image thresholding by minimizing the measures of fuzziness. *Pattern Recogn.* **28**, 41–51 (1995)
11. Qiao, Y., Hu, Q.M., Qian, G.Y., Luo, S.H., Nowinski, W.L.: Thresholding based on variance and intensity contrast. *Pattern. Recogn.* **40**, 596–608 (2007)
12. Li, X.Q., Zhao, Z.W., Cheng, H.D.: Fuzzy entropy threshold approach to breast cancer detection. *Inf. Sci. Appl.* **4**, 49–56 (1995)
13. Kittler, J., Illingworth, J.: Minimum error thresholding. *Pattern Recogn.* **19**, 41–47 (1986)
14. Li, C.H., Tam, P.K.S.: An iterative algorithm for minimum cross entropy thresholding. *Pattern Recogn. Lett.* **19**, 771–776 (1998)
15. de Albuquerque, M.P., Esquef, I.A., Mello, A.R.G.: Image thresholding using Tsallis entropy. *Pattern. Recogn. Lett.* **25**, 1059–1065 (2004)
16. Kurban, T., Civicioglu, P., Kurban, R., Besdok, E.: Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding. *Appl. Soft Comput.* **23**, 128–143 (2014)
17. Akay, B.: A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Appl. Soft. Comput.* **13**, 3066–3091 (2013)
18. Ali, M., Ahn, C.W., Pant, M.: Multi-level image thresholding by synergetic differential evolution. *Appl. Soft. Comput.* **17**, 1–11 (2014)
19. Chander, A., Chatterjee, A., Siarry, P.: A new social and momentum component adaptive PSO algorithm for image segmentation. *Expert. Syst. Appl.* **5**, 4998–500 (2011)
20. Coello, C.C., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary algorithms for solving multi-objective problems.* Springer Science & Business Media, Berlin (2007)
21. Hammouche, K., Diaf, M., Siarry, P.: A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Comput. Vis. Image Underst.* **109**, 163–175 (2008)
22. Tao, W.B., Tian, J.W., Liu, J.: Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recogn. Lett.* **24**, 3069–3078 (2003)
23. Yin, P.Y.: A fast scheme for optimal thresholding using genetic algorithms. *Signal Process.* **72**, 85–95 (1999)
24. Zhang, J., Percy, R.G., McCarty Jr., J.C.: Introgression genetics and breeding between Upland and Pima cotton: a review. *Euphytica* **198**, 1–12 (2014)
25. Ayala, H.V.H., dos Santos, F.M., Mariani, V.C., dos Santos Coelho, L.: Image thresholding segmentation based on a novel beta differential evolution approach. *Expert. Syst. Appl.* **42**, 2136–2142 (2015)
26. Sarkar, S., Das, S., Chaudhuri, S.S.: A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution. *Pattern Recogn. Lett.* **54**, 27–35 (2015)
27. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global. Optim.* **11**, 341–359 (1997)
28. Kirkpatrick, S.: Optimization by simulated annealing: quantitative studies. *J. Stat. Phys.* **34**, 975–986 (1984)
29. Drigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperation agents. *IEEE Trans. Syst. Man Cybern. B* **26**, 29–41 (1996)
30. Tao, W.B., Jin, H., Liu, L.M.: Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recogn. Lett.* **28**, 788–796 (2007)
31. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. In: Broy, M., Dener, E. (eds.) *Software Pioneers*, pp. 10–13. Springer, Heidelberg (2002)
32. Civicioglu, P.: Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput. Geosci.* **46**, 229–247 (2012)
33. Clerc, M., Kennedy, J.: The particle swarm—exploration, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **6**, 58–73 (2002)
34. Nabizadeh, S., Faez, K., Tavassoli, S., Rezvanian, A.: *A Novel Method for Multi-level Image Thresholding Using Particle Swarm Optimization Algorithms.* IEEE, New York (2010)

35. Yin, P.Y.: Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Appl. Math. Comput.* **184**, 503–513 (2007)
36. Baniani, E.A., Chalechale, A.: Hybrid PSO and genetic algorithm for multilevel maximum entropy criterion threshold selection. *Int. J. Hybrid Inf. Technol.* **6**, 131–140 (2013)
37. Juang, C.F.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *Proc IEEE Trans. Syst. Man Cybern. B* **34**, 997–1006 (2004)
38. Patel, M.K., Kabat, M.R., Tripathy, C.R.: A hybrid ACO/PSO based algorithm for QoS multicast routing problem. *Ain Shams Eng. J.* **5**, 113–120 (2014)
39. Zhang, Y.D., Wu, L.N.: A robust hybrid restarted simulated annealing particle swarm optimization technique. *Adv. Comput. Sci. Appl.* **1**, 5–8 (2012)
40. Boussaid, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M.: *Hybrid BBO-DE Algorithms for Fuzzy Entropy-based Thresholding*. Springer, Heidelberg (2013)
41. Gong, W.Y., Cai, Z.H., Ling, C.X., Li, H.: A real-coded biogeography-based optimization with mutation. *Appl. Math. Comput.* **216**, 2749–2758 (2010)
42. John, H.: *Holland. Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge (1992)
43. Li, C.H., Lee, C.: Minimum cross entropy thresholding. *Pattern Recogn.* **26**, 617–625 (1993)
44. Tsai, W.H.: Moment-preserving thresholding: a new approach. *Comput. Vision Graph.* **29**, 377–393 (1985)
45. Blum, C., Li, X.: *Swarm Intelligence in Optimization*. Springer, Heidelberg (2008)
46. Kennedy, J., Eberhart, R.C., Labahn, G.: *Particle Swarm Optimization*. Kluwer, Boston (1995)
47. Yang, X.S., Deb, S.: *Cuckoo Search via Lévy Flights*. IEEE, New York (2009)
48. Yang, X.S., Hossein, G.A.: Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29**, 464–483 (2012)
49. Mirjalili, S., Mirjalili, S.Z.M., Lewis, A.: Grey Wolf Optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
50. Gong Y.J., Li, J.J., Zhou, Y., Li, Y., Chung, H.S.H., Shi, Y.H., Zhang, J.: Genetic learning particle swarm optimization. *IEEE Trans. Cybern.*, 1–14 (2015)
51. Zhang, Y.D., Wu, L.N.: Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. *Entropy* **13**, 841–859 (2011)
52. Alihodzic, A., Tuba, M.: Bat algorithm (BA) for image thresholding. *Recent Res. Telecommun. Inf. Electron. Signal Process.*, 17–19 (2013)
53. Alihodzic, A., Tuba, M.: Improved bat algorithm applied to multilevel image thresholding. *Sci World J.* (2014). doi:[10.1155/2014/176718](https://doi.org/10.1155/2014/176718)
54. Ye, Z.W., Wang, M.W., Liu, W., Chen, S.B.: Fuzzy entropy based optimal thresholding using bat algorithm. *Appl. Soft. Comput.* **31**, 381–395 (2015)
55. Li, Y.Y., Jiao, L.C., Shang, R.H., Stolkin, R.: Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. *Inf. Sci.* **294**, 408–422 (2015)
56. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**, 2232–2248 (2009)
57. Birbil, S., Fang, S.C.: An electromagnetism-like mechanism for global optimization. *J. Global Optim.* **25**, 263–282 (2003)
58. Kaveh, A., Talatahari, S.: A novel heuristic optimization method: charged system search. *Acta. Mech.* **213**, 267–289 (2010)
59. Biswas, A., Mishra, K.K., Tiwari, S., Misra, A.K.: Physics-inspired optimization algorithms: a survey. *J. Optim.* **2013**, 1–16 (2013)
60. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comp.* **1**, 67–82 (1997)
61. Jiang, S., Wang, Y., Ji, Z.: Convergence analysis and performance of an improved gravitational search algorithm. *Appl. Soft Comput.* **24**, 363–384 (2014)
62. Kumar, J.V., Kumar, D.V., Edukondalu, K.: Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market. *Appl. Soft Comput.* **13**, 2445–2455 (2013)
63. Mirjalili, S., Hashim, S.Z.M., Sardroudi, H.M.: Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Appl. Math. Comput.* **218**, 11125–11137 (2012)

64. Sabri, N.M., Puteh, M., Mahmood, M.R.: A review of gravitational search algorithm. *Int. J. Adv. Soft Comput. Appl.* **5**, 1–39 (2013)
65. Sun, G.Y., Zhang, A.Z., Yao, Y.J., Wang, Z.J.: A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding. *Appl. Soft Comput.* **46**, 703–730 (2016). doi:[10.1016/j.asoc.2016.01.054](https://doi.org/10.1016/j.asoc.2016.01.054)
66. Mirjalili, S., Hashim, S.Z.M.: A new hybrid PSOGSA algorithm for function optimization. IEEE, New York (2010)
67. Mirjalili, S., Lewis, A.: Adaptive gbest-guided gravitational search algorithm. *Neural Comput. Appl.* **25**, 1569–1584 (2014)
68. Herrera, F., Lozano, M., Verdegay, J.L.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy. Set. Syst.* **92**, 21–30 (1997)
69. Sahoo, P.K., Soltani, S., Wong, A.K.C.: A survey of thresholding techniques. *Neural. Comput. Appl.* **41**, 233–260 (1988)
70. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **1**, 3–18 (2011)
71. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Ttans. Evolut. Comput.* **10**, 281–295 (2006)