Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding

Genyun Sun [a,*], Aizhu Zhang [a], Yanjuan Yao [b], Zhenjie Wang [a]

[a] School of Geosciences, China University of Petroleum (East China), Qingdao, Shandong 266580, China
[b] Satellite Environment Center (SEC), Ministry of Environmental Protection (MEP) of China, Beijing 100094, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Multi-level thresholding
Image segmentation
Genetic algorithm
Gravitational search algorithm
Entropy
Between-class variance

ABSTRACT

The multi-level thresholding is a popular method for image segmentation. However, the method is computationally expensive and suffers from premature convergence when level increases. To solve the two problems, this paper presents an advanced version of gravitational search algorithm (GSA), namely hybrid algorithm of GSA with genetic algorithm (GA) (GSA-GA) for multi-level thresholding. In GSA-GA, when premature convergence occurred, the roulette selection and discrete mutation operators of GA are introduced to diversify the population and escape from premature convergence. The introduction of these operators therefore promotes GSA-GA to perform faster and more accurate multi-level image thresholding. In this paper, two common criteria (1) entropy and (2) between-class variance were utilized as fitness functions. Experiments have been performed on six test images using various numbers of thresholds. The experimental results were compared with standard GSA and three state-of-art GSA variants. Comparison results showed that the GSA-GA produced superior or comparative segmentation accuracy in both entropy and between-class variance criteria. Moreover, the statistical significance test demonstrated that GSA-GA significantly reduce the computational complexity for all of the tested images.

1. Introduction

Image segmentation is an important technology for image processing and is a fundamental process in many image, video, and computer vision applications [1]. It is useful for separating objects from backgrounds or discriminating objects from objects with distinct gray-levels. Image thresholding is often used in image segmentation. The main purpose of image segmentation is to determine an efficient threshold (for bi-level thresholding) or several thresholds (for multi-level thresholding) according to some criteria [2]. Bi-level thresholding classifies pixels into two groups: one group includes pixels with gray-levels above a certain threshold, and the other group includes the rest. This is the simplest thresholding problem, as only one gray value need to be found. However, for real-world image processing, multi-level thresholding is more often utilized. The multi-level thresholding method divides pixels into several groups by assigning a single intensity value to pixels that belong to a group. In each group, pixels have intensity values within a specific range.

Multi-level thresholding techniques can be classified into two types: optimal thresholding methods and property-based thresholding methods [3]. Optimal thresholding methods search for the optimal thresholds by optimizing an objective function based on image gray-level histogram. Researchers have proposed several algorithms to determine objective functions, where thresholds are determined by jointly maximizing class uncertainty and region homogeneity. The algorithms include maximizing entropy to measure the homogeneity of segmented classes (e.g. Kapur's entropy) [4,5], maximizing the separability measure based on between-class variance (e.g. Otsu method) [6,7], thresholding based on the fuzzy similarity measure [8], and minimizing of Bayesian error [9], etc. Among them, the Otsu method [6] and the Kapur's entropy method [4] are the most preferred methods and they are relatively easy to use [10]. Therefore, they are selected as the objective functions in this study. However, with the thresholds increase, all these methods encounter two common problems: (1) the computational complexity increases as exponentially and (2) prone to premature convergence [7,11,12].

Over the past decades, researchers have developed several algorithms to solve these types of optimization problems, including branch-and-bound [13], meta-heuristics [14], and gradient-based methods [15]. Among them, meta-heuristics based methods have been extensively employed for performing fast search of optimal thresholds because of their significantly advantages in escaping from local optima. The popular meta-heuristic algorithms include genetic algorithm (GA) [6], simulated annealing (SA) [16], ant colony optimization (ACO) [17], artificial bee colony optimization (ABC) [18], differential evolution (DE) [19], biogeography-based optimization (BBO) [20], differential search algorithm (DS) [21], particle swarm optimization (PSO) [22], and so on. Moreover, to further improve the convergence accuracy and speed, a number of hybrid meta-heuristic algorithms have been proposed, such as GAPSO (hybrid GA with PSO) [23,24], ACO/PSO (hybrid ACO with PSO) [25], SA/PSO (hybrid SA with PSO) [26], BBO-DE (hybrid DE with BBO) [20], etc. Recently, many of these algorithms and their variants have been applied to multi-level thresholding as illustrated in Table 1.

Generally speaking, all these pure meta-algorithms have achieved certain successes and have showed different advantages. For example, the DE is the most efficient with respect to the quality of the optimal thresholds compared with GA, PSO, ACO, and SA whereas PSO converges the most quickly when comparing with ACO, GA, DE, and SA [29]. Besides, the DS consumes the shortest running time for multi-level color image thresholding when comparing with DE, GA, PSO, ABC, etc. [10].

* Corresponding author.
E-mail address: genyunsun@163.com (G. Sun).
URL: http://geori.upc.edu.cn/photo/html/?174.html (G. Sun).

http://dx.doi.org/10.1016/j.asoc.2016.01.054
1568-4946/

Table 1
Meta-heuristic algorithms for multi-level thresholding.

Meta-heuristic algorithms	Author	Year	Ref.
GA	Yin	1999	[27]
	Tao, Tian, and Liu	2003	[28]
	Hammouche, Diaf, and Siarry	2008	[29]
PSO	Clerc and Kennedy	2002	[30]
	Yin	2007	[31]
	Nabizadeh, Faez, and Tavassoli et al.	2010	[32]
	Akay	2013	[7]
DE	Ali, Ahn, and Pant	2014	[12]
	Ayala, dos Santos, and Mariani et al.	2015	[33]
	Sarkar, Das, and Chaudhuri	2015	[5]
ACO	Yin	2007	[31]
ABC	Akay	2013	[7]
SA	Kurban, Civicioglu, and Kurban et al.	2014	[10]
DS	Kurban, Civicioglu, and Kurban et al.	2014	[10]
GAPSO	Juang	2004	[24]
	Baniani and Chalechale	2013	[23]
SA/PSO	Zhang and Yu	2012	[26]
BBO-DE	Boussaï d, Chatterjee, and Siarry et al.	2013	[20]

Nevertheless, the deficiencies of the algorithms themselves still weaken their application in multi-level thresholding. For example, in PSO, a particle i can only learn from the experience of its neighbors (donated by ***gbest***) and the experience of itself (donated by $\boldsymbol{pbest}_i$), if the ***gbest*** is trapped, the convergence process will suffer from premature convergence [34,35]. The success of DE in solving a specific problem crucially depends on appropriately choosing trial vector generation strategies and their associated control parameter values [36]. Therefore, a number of hybrid meta-heuristic algorithms have been presented for multi-level thresholding, in which the GAPSO [23] and BBO-DE [20] have been proved to be preferable algorithms. However, GAPSO cannot obtain high-quality optimal thresholding sometimes for its performance heavily depends on the settings of three parameters and the topology structure of the neighbors [37]. Similarly, BBO-DE performs better than the pure DE, but its optimization capability still crucially depends on the selecting of trial vector generation strategies and their associated control parameter values.

In short, the unavoidable disadvantages of the meta-heuristic algorithms make it still a challenge task to obtain the optimal thresholds rapidly while maintaining high quality capabilities [12]. Consequently, many researches have been focusing on improving the existing algorithms as well as exploiting new meta-heuristic algorithms.

As one of the newest meat-heuristic algorithms, the gravitational search algorithm (GSA), which is inspired by the law of gravity and mass interactions, has proven its promising efficiency for solving complex problems [38]. Compared to the aforementioned meta-heuristic algorithms, GSA possesses simpler concept, higher computational efficiency, and fewer parameters [39]. A number of researches have reported the superiority of GSA in terms of the convergence precision, convergence speed, and stable convergence characteristics over many other meta-heuristic algorithms, such as PSO, GA, Central Force Optimization (CFO), and ACO [38,40–43], etc. These advantages of GSA make it a potential choice for solving multi-level thresholding. However, due to the gravitational force absorbs masses into each other, no recovery for GSA if premature convergence occurs [44]. In this situation, GSA loses its ability of explore and the convergence speed in the last iterations is slow. New operators should thus be added to GSA to prevent premature convergence and to increase its flexibility in solving more complicated problems [44].

In the past few years, many researches have paid close attention to the improvement of GSA and presented some GSA variants, such as [39–42,45–47]. Most of the GSA variants were presented to prevent the premature convergence problem or decrease the computational complexity of GSA by designing new learning strategies or hybrid with other algorithms/operators. However, very few of the algorithms have focused on the application on multi-level thresholding. When applying GSA into multi-level thresholding, especially when the number of thresholds is increased, the two problems, premature convergence and high computational complexity, become more serious.

Actually, the lack of diversity is one important reason for the premature convergence [48]. As basic operators to provide the necessary diversity within a population, mutations have been utilized in many meta-heuristic optimization algorithms [49,50]. GA, as an adaptive meta-heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics, is famous for its mutation operator [51]. Keeping this in view, the population diversity in GSA can be greatly enhanced by the hybrid of GSA with GA. However, due to the rotatory hybrid may cause high computational complexity, taking the evolutionary stages into consideration to guide the hybrid of GSA and GA is necessary.

Based on the above analysis, in this paper, we developed a novel hybrid algorithm of GSA with GA (GSA-GA) for multi-level thresholding. In GSA-GA, the discrete mutation operator of GA was introduced to promote the population diversity when premature convergence occurred. To identify whether the population suffers from premature convergence, we calculated the standard deviation of objective functions first. If the standard deviation is smaller than a random number $rand \in [0, 1]$ or the $rand$ is bigger than the ratio of the current iteration t to the maximum iterations $Iter_{max}$, the mutation operator is carried out. This makes the GSA-GSA is provided with adaptive characteristics along with the evolutionary stages. Moreover, for selecting the particles for mutation, the roulette selection of GSA was also introduced. The introduction of these operators therefore could promote GSA-GA to perform faster and more accurate multi-level image thresholding. The entropy and between-class variance were respectively considered as evaluation criteria for GSA-GA.

The remainder of this paper is organized as follows. Section 2 first briefly describes the frameworks of GA and GSA, and then reviews the entropy and between-class variance criteria. In Section 3, details of the proposed GSA-GA are given followed by the implement of GSA-GA for multi-level thresholding. The experimental set up and results are included in Section 4. Finally, the paper is concluded in Section 5.

2. Background

2.1. GA

GA is a stochastic search algorithm based on the mechanics of natural evolution which can be used to solve optimization problems [51,52]. It starts optimization with a randomly initialized population $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ where N is the size of the population. In the population, each individual $\boldsymbol{x}_i$ represents a point in the search space and thus a candidate solution to the problem. Each candidate solution for a specific problem is called a chromosome and contains a linear list of genes. GA then uses three basic operators (selection, crossover, and mutation) to manipulate the genetic composition of a population. Selection is a process in which individuals with the highest fitness values in the current generation are reproduced in the new generation. The crossover operator produces two offspring (new candidate solutions) by recombining the information from two parents. Mutation is a random alteration of some gene values in an individual. The allele of each gene is a candidate for mutation, and the mutation probability determines its function [24]. In the new generation, the population is more adapted to the environment than the previous generation, and the evolution continues until meeting an optimization criterion. After decoding the last individual, an optimal solution can be gained.

2.2. GSA

In GSA, a particle $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iS}]$ moves through the S-dimensional search space with the velocity $\boldsymbol{v}_i = [v_{i1}, v_{i2}, \ldots, v_{iS}]$

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

which is determined by the gravitational forces exerted by its neighbors according to the law of gravity [38]. Due to the force between two particles is directly proportional to their masses and inversely proportional to their distance, all the particles move towards those particles that have heavier masses [38]. The mass of each particle in generation t, denoted by $Mass_i^t$, is simply calculated by Eqs. (1) and (2) as follows:

$$nmfit_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t}, \tag{1}$$

$$Mass_i^t = \frac{nmfit_i^t}{\sum_{j=1}^{N} nmfit_j^t}, \tag{2}$$

where fit_i^t represents the fitness value of the particle i at time t. For a maximization problem, $worst^t$ and $best^t$ are defined in Eqs. (3) and (4) as follows:

$$worst^t = \min_{j \in [j=1,2,\ldots,N]} fit_j^t, \tag{3}$$

$$best^t = \max_{j \in [j=1,2,\ldots,N]} fit_j^t. \tag{4}$$

In an optimization problem, the force acting on the particle i from the particle j at a specific time t is shown in Eq. (5) as follows:

$$F_{ij}^d(t) = G(t)\frac{Mass_i^t \times Mass_j^t}{R_{ij}^t + \varepsilon}(x_{jd}^t - x_{id}^t), \tag{5}$$

where x_{id}^t represents the position of the i-th particle in the d-th dimension; x_{jd}^t represents the position of the j-th particle in the d-th dimension; $Mass_i^t$ and $Mass_j^t$ are the gravitational mass related to the particle i and j; $G(t)$ is the gravitational constant; R_{ij}^t is the Euclidian distance between particles i and j in generation t; ε is a small constant which is bigger than 0.

To give a stochastic characteristic to GSA, the total force that acts on the particle i in the d-th dimension is set to be a randomly weighted sum of d-th components of the forces exerted from other particles as shown in Eq. (6) as follows:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{N} rand \cdot F_{ij}^d(t), \tag{6}$$

where N is the size of the population, and $rand$ is a uniform random variable in the interval [0, 1].

Hence, by the law of motion, the acceleration of the particle i at time t, and in the d-th dimension, a_{id}^t, is given in Eq. (7) as follows:

$$a_{id}^t = \frac{F_i^d(t)}{Mass_i^t}. \tag{7}$$

The gravitational constant, G, is initialized to G_0 at the beginning and decreases with time to control the search accuracy. It is defined in Eq. (8) as follows:

$$G(t) = G_0 \times \exp\left(-\beta \times \frac{t}{Iter_{\max}}\right), \tag{8}$$

where β is the coefficient of decrease, t is the current generation, and $Iter_{\max}$ is the maximum number of iterations. In the standard GSA, G_0 is set to 100 and β is set to 20. This setting is adopted by all the GSA variants of this paper.

In generation t, the velocity and the position of the particle i are updated according to Eqs. (9) and (10) as follows:

$$v_{id}^{t+1} = rand \times v_{id}^t + a_{id}^t, \tag{9}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \tag{10}$$

where $rand$ is a uniform random variable in the interval [0, 1].

2.3. Entropy criterion

The entropy criterion, proposed by Kapur et al. [4], is original developed for bi-level thresholding. The entropy is maximum when the optimal thresholds separating the image properly. The entropy of an image is often obtained from its probability distribution. Assume there are L gray-levels in a given image and the gray-levels are in the range $\{0, 1, 2, \ldots, (L-1)\}$. One can define probability of the gray-level i by Eq. (11) as follows:

$$prob_i = \frac{hist_i}{\sum_{i=0}^{L-1} hist_i}, \tag{11}$$

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

For bi-level thresholding, entropy can be described by Eq. (12) as follows:

$$\begin{aligned} HE_0 &= -\sum_{i=0}^{td_0-1} \frac{prob_i}{\omega_0} \ln \frac{prob_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{td_0-1} prob_i \\ HE_1 &= -\sum_{i=td_0}^{L-1} \frac{prob_i}{\omega_1} \ln \frac{prob_i}{\omega_1}, \quad \omega_1 = \sum_{i=td_0}^{L-1} prob_i \end{aligned} \tag{12}$$

where HE_i ($i = 0, 1$) is the entropy of each class, td_0 is the optimum threshold of the bi-level thresholding, and ω_i ($i = 0, 1$) is the probability of each class.

The optimum threshold td_0 can be used to maximizes the fitness function Eq. (13) as follows:

$$f(td_0) = HE_0 + HE_1, \tag{13}$$

This method has been extended to solve multi-level thresholding problems. Assume a problem of determining m thresholds for a given image $[td_1, td_2, \ldots, td_m]$. The objective is to maximize the fitness function Eq. (14) as follows:

$$f([td_1, td_2, \ldots, td_m]) = HE_0 + HE_1 + HE_2 + \cdots + HE_m, \tag{14}$$

where

$$\begin{aligned} H_E0 &= -\sum_{i=0}^{td_1-1} \frac{prob_i}{\omega_0} \ln \frac{prob_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{td_1-1} prob_i \\ HE_1 &= -\sum_{i=td_1}^{td_2-1} \frac{prob_i}{\omega_1} \ln \frac{prob_i}{\omega_1}, \quad \omega_1 = \sum_{i=td_1}^{td_2-1} prob_i \\ HE_2 &= -\sum_{i=td_2}^{td_3-1} \frac{prob_i}{\omega_2} \ln \frac{prob_i}{\omega_2}, \quad \omega_2 = \sum_{i=td_2}^{td_3-1} prob_i \\ HE_m &= -\sum_{i=td_m}^{L-1} \frac{prob_i}{\omega_m} \ln \frac{prob_i}{\omega_m}, \quad \omega_m = \sum_{i=td_m}^{L-1} prob_i \end{aligned} \tag{15}$$

2.4. Between-class variance criterion

Thresholding based on between-class variance criterion is another histogram-based image segmentation algorithm. The methods divided the image into classes so that the total variance of different classes is maximized. Otsu [6] defined the object function Eq. (16) as the sum of sigma functions of each class:

$$f(td_0) = \sigma_0 + \sigma_1, \tag{16}$$

The sigma functions are obtained from Eq. (17) as follows:

$$\begin{aligned} \sigma_0 &= \omega_0(\mu_0 - \mu_T)^2, \quad \mu_0 = \frac{\sum_{i=0}^{td_0-1} i \cdot prob_i}{\omega_0}, \\ \sigma_1 &= \omega_1(\mu_1 - \mu_T)^2, \quad \mu_1 = \frac{\sum_{i=td_0}^{L-1} i \cdot prob_i}{\omega_1}, \end{aligned} \tag{17}$$

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

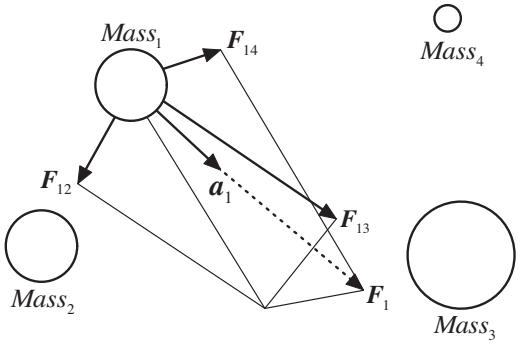


Fig. 1. Schematic diagram of particle's movement in GSA.

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

Similar to the entropy method, the between-class variance criterion was extent to multi-level thresholding problems. The objective is to maximize the following function Eq. (18):

$$f([td_1, td_2, \ldots, td_m]) = \sigma_0 + \sigma_1 + \sigma_2 + \cdots + \sigma_m, \tag{18}$$

where

$$\begin{aligned}
\sigma_0 &= \omega_0(\mu_0 - \mu_T)^2, \quad \mu_0 = \frac{\sum_{i=0}^{td_1-1} i \cdot prob_i}{\omega_0}, \\
\sigma_1 &= \omega_1(\mu_1 - \mu_T)^2, \quad \mu_1 = \frac{\sum_{i=td_1}^{td_2-1} i \cdot prob_i}{\omega_1}, \\
\sigma_2 &= \omega_2(\mu_2 - \mu_T)^2, \quad \mu_2 = \frac{\sum_{i=td_2}^{td_3-1} i \cdot prob_i}{\omega_2}, \\
\sigma_m &= \omega_m(\mu_m - \mu_T)^2, \quad \mu_m = \frac{\sum_{i=td_m}^{L-1} i \cdot prob_i}{\omega_m},
\end{aligned} \tag{19}$$

3. GSA-GA based multi-level thresholding image segmentation

In the standard GSA, the movement of particle is leaded by the acceleration as illustrated in Eqs. (9) and (10). In other words, the search direction of each particle is determined by the resultant force exerted by all its neighbors in current time as depicted in Fig. 1.

Although the resultant force guidance property makes GSA convergence fast in the early iterations, it causes GSA susceptible to local optima [44]. GSA thus easily suffers from premature convergence especially for complex problems. Once premature convergence occurs, the population diversity is poor and GSA loses its ability to explore. Moreover, GSA performs slow convergence speed in the last iterations for premature convergence can result in vibration problem [41].

3.1. GSA-GA

To tackle the aforementioned problems, we proposed a novel hybrid algorithm of GSA with GA (GSA-GA) in this paper. In GSA-GA, when particles are trapped, the roulette selection [53] and discrete mutation [54] operators of GA are introduced. In this study, to adaptively identify whether the hybrid of GSA and GA is necessary, the standard deviations of objective functions and the ratio of the current iteration t to the maximum iterations $Iter_{\max}$ are calculated before conducting the roulette selection and discrete mutation of GA.

More specifically, when particles are trapped, the fitness values (***fit***) of different particles become extraordinary similar to each other. Thus, the current population lacks of population diversity and the fitness value standard deviation *std(**fit**)* is very small. To diversify the population, if *rand<std(**fit**)*, the roulette selection and discrete mutation operators are carried out to update population. Due to those better particles are given more opportunities to be chosen in the roulette selection, its combination with mutation can help GSA-GA escape from local optima.

Besides, if $rand<t/Iter_{\max}$, the roulette selection and discrete mutation operators are also performed. Apparently, satisfaction of the condition '$rand<t/Iter_{\max}$' becomes easier by the lapse of time. Hence, GSA-GA can utilize selection and mutation operators to accelerate convergence in the last iterations. The principle of GSA-GA is shown in Fig. 2.

In the roulette selection, selection probability of a particle is in proportion to the quality of its original fitness [53]. For a maximum problem, a particle with bigger objective function value is more likely selected. Conversely, a particle with smaller objective function value is more likely selected in a minimum problem. Take maximum optimization problem as example, the selection probability of the particle i in generation t is calculated by Eq. (20) as follows:

$$p_s^t(i) = \frac{fit_i^t}{\sum_{j=1}^{N} fit_i^t}, \tag{20}$$

The cumulative probability of the particle i in generation t thus can be calculated by Eq. (21) as follows:

$$p_{cs}^t(i) = \sum_{j=1}^{i} p_s^t(i), \tag{21}$$

Then, generate a value *rand* randomly between 0 and 1 for each particle, and find the sequence number j in the array of cumulative probability with Eq. (22) as follows:

$$p_{cs}^t(j) < rand < p_{cs}^t(j+1), \quad j \in [1, 2, \ldots, N], \tag{22}$$

Particle j therefore can be selected to the next generation as shown in Eq. (23) as follows:

$$\boldsymbol{Newx}_i^t = \boldsymbol{x}_j^t, \tag{23}$$

Apparently, the roulette selection gives those better particles more opportunities to be chosen for mutation. Then, the discrete mutation operator and real-coded were implemented. Following a mutation rate p_m, the discrete mutation is performed by Eq. (24) as follows:

$$\boldsymbol{x}_i^{t+1} = rem(\boldsymbol{Newx}_i^t + (rand < p_m). * ceil(rand. * (\boldsymbol{BaseM} - 1)), \boldsymbol{BaseM}), \tag{24}$$

where ***BaseM*** is a mutation mask matrix calculated based on the search space.

Take the optimization of function Sphere [38] as example. In the final iterations, before the implement of GA operators, particles are easy to trap in a small range as shown in Fig. 3(a). In contrast, after performing the two GA operators, particles search more broadly as shown in Fig. 3(b). Therefore, with the roulette selection and discrete mutation operators, GSA-GA diversify the population and can escape from local optima.

3.2. Implement of GSA-GA for multi-level thresholding

The application of GSA-GA approach to the multi-level image thresholding problem depends on the criterion used for optimization. In this paper, both the entropy and between-class variance criteria were implemented. To start the GSA-GA for multi-level thresholding, initial population $\boldsymbol{x}$ should be randomly generated first. Each particle is a candidate solution of the required thresholds. The size of the population N is set by users, and dimension

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

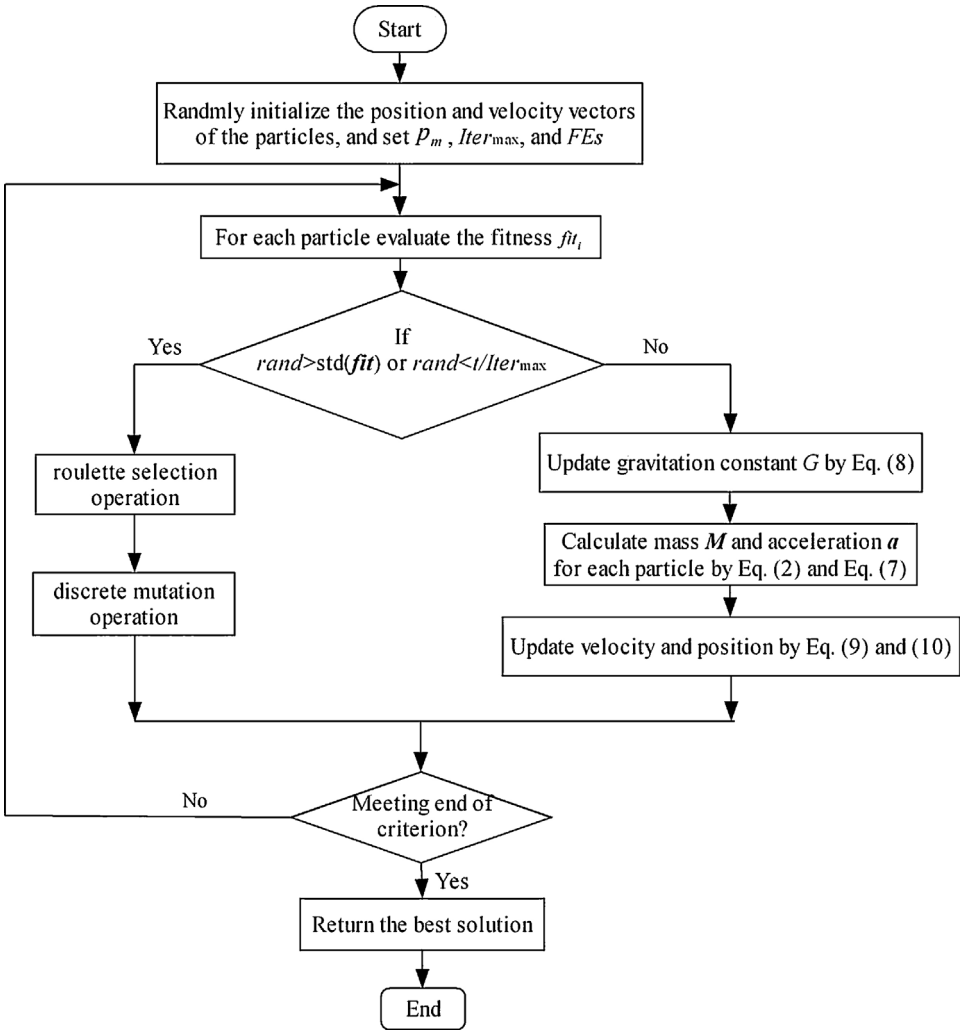


Fig. 2. The GSA-GA principle.

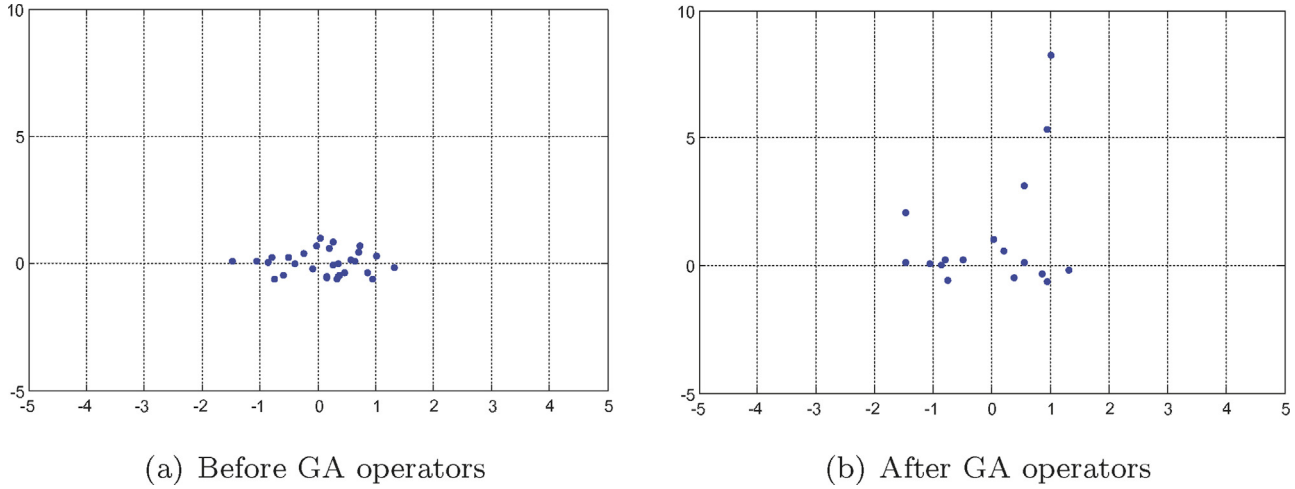


Fig. 3. The position of the particles observed before and after the GA operators in GSA-GA process for Sphere function, $S = 2$, and $N = 30$.

S of each particle is the number of thresholds: $S = m$. In the iteration process, the fitness value of each particle is calculated from the entropy or between-class variance criterion using Eqs. (14) and (18), respectively. The pseudocode of the GSA-GA algorithm for multi-level image thresholding is shown in Fig. 4.

4. Experiments and results

To demonstrate the validity of the proposed algorithm, we compared GSA-GA with the standard GSA [38], GGSA (Adaptive gbest-guided GSA) [46], PSOGSA (hybrid PSO and GSA) [45], DS [21],

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

1. Input test image ***IMG***.
2. Set p_m, $Iter_{\max}$, N, G_0, β, m, and FEs.
3. Set the thresholding criterion F_{index}='entropy' or 'between-class variance'.
4. Initialize positions $\boldsymbol{x}$ and velocities $\boldsymbol{v}$ of particles in the population.
5. **For** $t=1:Iter_{\max}$
6. **If** F_{index}='entropy'
7. Evalutate the fitness functions values ***fit*** by Eq. (14);
8. **Else**
9. Evalutate the fitness functions values ***fit*** by Eq. (18);
10. **End**
11. **End**
12. **If** $(rand>std\ (\boldsymbol{fit})\ \|\ rand<(t/Iter_{\max})$
13. Calculate the selection probability $p_s^t(i)$ of each particle i by Eq.(20);
14. Calculate cumulative probability $p_{cs}^t(i)$ of each particle i by Eq.(21);
15. **For** $i=1:N$
16. Generate $rand$ randomly;
17. **If** $p_{cs}^t(j) < rand < p_{cs}^t(j+1) \quad j \in [1,2,\ldots,N]$
18. $\boldsymbol{Newx}_i^t = \boldsymbol{x}_j^t$;
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 $\boldsymbol{BaseM} = \boldsymbol{BaseV}(ones(N,1),:)$;
23. $\boldsymbol{x}^{t+1} = rem(\boldsymbol{Newx}^t + (rand < p_m).*ceil(rand.*(BaseM-1)), BaseM)$;
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 $Mass_i^t = \frac{nmfit_i^t}{\sum_{j=1}^{N} nmfit_j^t}$ and $nmfit_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t}$;
28. Calculate the force acting on particle i from other parrticles in demension d
29. by $F_{ij}^d(t) = G(t)\frac{Mass_i^t \times Mass_j^t}{R_{ij}^t + \varepsilon}(x_{jd}^t - x_{id}^t)$ and $F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand \cdot F_{ij}^d(t)$;
30. Calculate the acceleration by $a_{id}^t = \frac{F_i^d(t)}{Mass_i^t}$;
31. Update the position and velocity of particle i by:
32. $v_{id}^{t+1} = rand \times v_{id}^t + a_{id}^t$;
33. $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$;
34. **End**
35. **End**
36. **End**

Fig. 4. Pseudocode of the GSA-GA algorithm for multi-level image thresholding.

BBO-DE [20], and GAPSO [23] by conducting experiments on 6 well-known standard benchmark images. In these compared algorithms, the GGSA and PSOGSA are two state-of-art GSA variants, the DS is one of the newest high-efficient meta-heuristic algorithms, and the BBO-DE, GAPSO are two kinds of new hybrid meta-heuristic algorithms which have been successfully utilized to solve multi-level thresholding problem. The test images are given in Fig. 5 as described at Section 4.1. Subsequently, the related experimental settings are presented in Section 4.2. In Section 4.3, the performance metrics are briefly introduced. Afterwards, in Sections 4.4 and 4.5, the multi-level image thresholding results using GSA-GA with between-class variance and entropy criteria respectively are compared to the six comparison algorithms. In these experiments, the number of thresholds in the range of 2-5 for each image is tested. Finally, a detailed analysis of the running time is presented in Section 4.6. To reduce statistical errors, each test is repeated 30 times independently.

4.1. Test images

In this paper, 6 well-known standard benchmark images, including Lena, Cameraman, butterfly, Landscape, Starfish, and House, are utilized to examine the performance of GSA-GA as shown in Fig. 5. The first three test images (Fig. 5(a) Lena image, Fig. 5(b) Cameraman image, and Fig. 5(c) House image) are of size 256×256 pixels with 8 bit gray-levels. The other three images are with size of 321×481 with 8 bit gray-levels. In addition, the first two images, Lena and Cameraman are taken from literature [55]. The House image, as shown in Fig. 5(c) is adopted from [56]. The rest three natural images, are taken form the Berkeley Segmentation Dataset [57].

4.2. Experimental settings

The relative parameter settings in this paper are shown in Table 2. As illustrated in Table 2, to perform a fair experiment, in all of the algorithms, the population size (N), and maximum number of fitness evaluations (FEs) were set to 30 and 3000, respectively. For GSA-GA, GSA, PSOGSA, and GGSA, the initial gravitational constant G_0 and decrease coefficient β were set to 100 and 20 as the default values in original GSA [38]. For PSOGSA, the two acceleration coefficients (c_1 and c_2) were set to 0.5 and 1.5, respectively as suggested in [45]. For GGSA, the two acceleration coefficients (c_1 and c_2) were set to $-2t^3/Iter_{\max}{}^3 + 2$ and $2t^3/Iter_{\max}{}^3$, respectively as recommended in [46]. The other parameters in DS, BBO-DE, and GAPSO were all adopted as the suggested values in the original

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

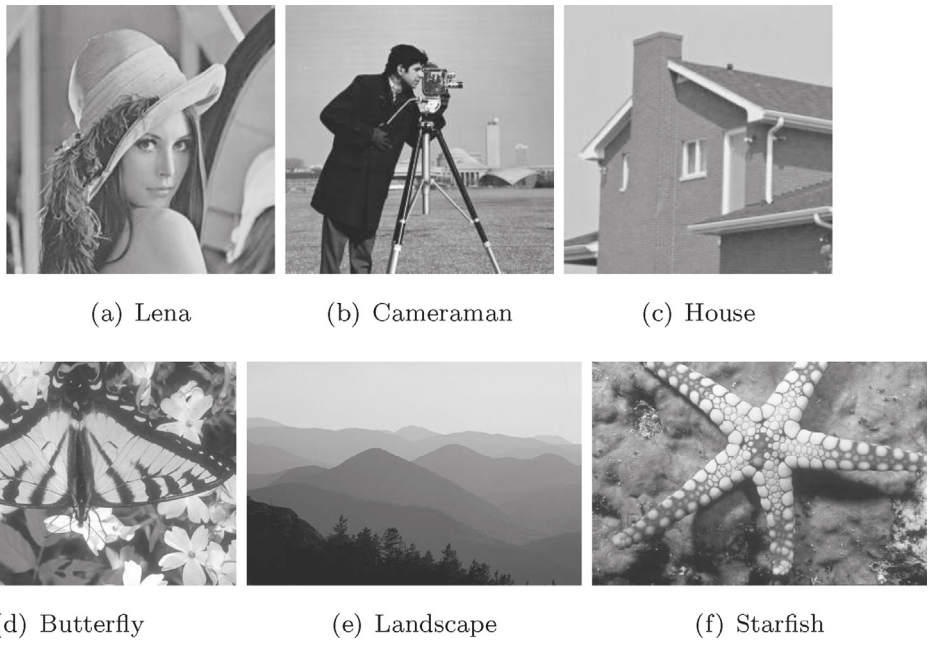


Fig. 5. Test images used in the experiments.

Table 2
Parameter settings in this paper.

parameters	GSA-GA	GSA	PSOGSA	GGSA	DS	BBO-DE	GAPSO
FEs	3000	3000	3000	3000	3000	3000	3000
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	–	–	–
β	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	–

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 (denoted by *CR*) was set to 0.9, the scaling factor (*F*) was set to 0.5, and the elitism parameter (denoted by n_{elit}) was set to 2 [20]; 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 [23]. For GSA-GA, the mutation probability p_m is set to 0.1 as it is recommended in many GA based algorithms [20,24,27].

4.3. Performance metrics

To qualitatively judge the segmentation procedure that produced thresholded images, we employed the popular uniformity measure [58], which has also been employed by Yin [27] to provide an effective performance comparison. The uniformity measure (u) is calculated by Eq. (25) as follows:

$$u = 1 - 2 \times m \times \frac{\sum_{j=1}^{m+1} \sum_{i \in Re_j} (f_i - g_j)^2}{Num \times (f_{\max} - f_{\min})^2}, \tag{25}$$

where m is the number of 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.

4.4. Experiment 1: Maximizing entropy

In this section, the entropy criterion was first maximized to construct an optimally segmented image for all the four GSA variants to confirm the superiority of GSA-GA over the original GSA and the two state-of-art GSA variants. Table 3 presented the best uniformity (u) and the corresponding threshold values (*Th*) on all the test images produced by the algorithms after 30 runs. To test the stability of the algorithms, Table 4 presented the mean and stand of deviation uniformity (*M_u* and *Std_u*). Moreover, to test the computation complexity, the mean and stand deviation of CPU times (*M_t* and *Std_t*) after 30 runs were reported in Table 5.

As shown in Table 3, GSA-GA performed better than the other GSA-based segmentation methods in most cases. Especially for the test image House, when $m = 3$, the proposed GSA-GA obviously outperformed the other three algorithms for the reason that GA's operators prevent GSA from getting stuck to local optima. Meanwhile, GSA-GA yielded the best *M_u* in most cases as illustrated by Table 4. The PSOGSA performed the best *Std_u* followed by GSA,

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 3
Comparison of algorithms (GSA, PSOGSA, GGSA and GSA-GA) taking the entropy as evaluation criteria in terms of best uniformity and corresponding thresholds. The best results of each metric are shown in boldface.

Image	m	GSA		PSOGSA		GGSA		GSA-GA	
		Th	u	*Th*	u	*Th*	u	*Th*	*u*
Lena	2	90,156	0.9688	92,158	0.9687	92,158	0.9687	97,157	**0.9690**
	3	76,120,169	**0.9773**	76,120,170	0.9771	76,120,169	**0.9773**	76,119,160	**0.9773**
	4	59,92,130,170	0.9789	59,92,131,171	0.9788	59,92,131,171	0.9788	69,107,145,170	**0.9809**
	5	53,81,108,144,177	0.9823	57,86,119,153,183	0.9813	56,84,115,144,175	**0.9832**	59,89,127,161,184	0.9800
Cameraman	2	128,193	0.9346	128,193	0.9346	128,193	0.9346	113,193	**0.9546**
	3	51,110,197	0.9665	98,146,197	0.9727	93,145,197	0.9681	48,113,199	**0.9758**
	4	47,97,147,193	**0.9826**	44,97,146,197	**0.9826**	44,96,146,193	**0.9826**	46,97,143,191	0.9824
	5	26,66,103,144,193	0.9817	25,62,100,146,197	0.9813	39,83,119,154,193	**0.9854**	40,82,115,146,194	0.9836
Butterfly	2	115,175	0.9688	115,176	0.9688	115,173	0.9688	98,176	**0.9738**
	3	76,119,177	0.9764	74,120,176	0.9764	74,120,176	0.9764	74,117,171	**0.9769**
	4	82,123,171,215	0.9797	74,120,172,219	0.9792	74,120,172,219	0.9792	77,123,167,208	**0.9803**
	5	64,96,134,173,217	**0.9815**	68,99,133,174,219	0.9808	68,97,128,173,218	0.9809	52,84,131,174,219	0.9808
Landscape	2	98,150	0.9443	98,150	0.9443	98,150	0.9443	96,157	**0.9499**
	3	81,124,191	0.9593	78,127,197	0.9591	77,126,197	0.9593	74,133,198	**0.9602**
	4	68,99,142,193	**0.9726**	28,59,100,151	0.9609	70,97,132,197	0.9660	72,98,131,188	0.9621
	5	23,60,97,132,175	**0.9847**	28,59,98,133,197	0.9834	28,59,97,129,171	0.9840	48,93,124,169,195	0.9832
Starfish	2	89,168	0.9669	91,170	0.9667	91,170	0.9667	92,161	**0.9671**
	3	72,124,180	0.9751	75,130,183	0.9741	74,129,183	0.9744	68,120,178	**0.9753**
	4	61,104,153,200	0.9769	68,116,164,206	0.9762	65,118,165,206	0.9748	65,110,155,201	**0.9771**
	5	54,90,129,168,208	0.9798	56,93,132,170,209	0.9796	54,90,127,164,206	**0.9805**	43,80,114,154,198	0.9804
House	2	103,197	0.9302	98,194	0.9265	98,193	0.9268	111,194	**0.9354**
	3	64,116,206	0.9146	65,112,194	0.9109	65,112,193	0.9113	64,133,195	**0.9630**
	4	59,97,132,193	0.9728	59,88,115,194	0.8932	61,101,137, 183	**0.9844**	52,82,145,197	0.9786
	5	66,101,134,164,201	0.9886	61,101,137,183,195	0.9868	61,101,137,182,194	0.9869	70,103,139,177,202	**0.9887**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 4
Comparison of algorithms (GSA, PSOGSA, GGSA and GSA-GA) taking the entropy as evaluation criteria in terms of the mean and standard deviation of uniformity. The best results of each metric are shown in boldface.

Image	*m*	GSA		PSOGSA		GGSA		GSA-GA	
		M_u	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*
Lena	2	**0.9687**	9.3439e−05	**0.9687**	**1.2413e−16**	**0.9687**	6.7388e−05	**0.9687**	9.0735e−04
	3	0.9754	2.8593e−04	**0.9771**	**0**	0.9745	7.2380e−05	**0.9771**	7.5622e−04
	4	0.9785	4.0528e−04	0.9787	**1.5286e−04**	0.9787	**1.5286e−04**	**0.9791**	0.0020
	5	0.9796	0.0024	0.9805	**4.7394e−04**	**0.9821**	6.0637e−04	0.9773	0.0034
Cameraman	2	0.9320	0.0053	0.9346	**0**	0.9346	**0**	**0.9355**	0.0135
	3	0.9635	**0.0017**	0.9650	0.0043	0.9655	0.0059	**0.9694**	0.0035
	4	0.9824	1.8244e−04	**0.9826**	**0**	0.9825	1.4102e−04	0.9814	0.0026
	5	0.9811	4.9037e−04	0.9813	**3.8962e−05**	**0.9845**	0.0018	0.9816	0.0037
Butterfly	2	0.9686	2.5721e−04	0.9688	**1.2413e−16**	0.9688	1.9051e−05	**0.9701**	0.0024
	3	0.9762	**4.5818e−04**	0.9704	0.0082	0.9733	0.0066	**0.9768**	0.0068
	4	0.9776	0.0053	**0.9792**	**0**	0.9785	0.0011	**0.9785**	7.2751e−04
	5	0.9800	0.0018	0.9791	**9.2801e−04**	0.9804	9.5647e−04	**0.9805**	0.0022
Landscape	2	0.9443	**0**	0.9443	**0**	0.9444	3.0594e−04	**0.9459**	0.0037
	3	0.9492	0.0134	0.9436	0.0087	**0.9592**	**9.2976e−05**	0.9550	0.0090
	4	0.9541	0.0119	0.9508	**0.0057**	**0.9565**	0.0077	0.9556	0.0181
	5	0.9801	**0.0061**	0.9727	0.0132	0.9769	0.0146	**0.9808**	0.0065
Starfish	2	0.9667	1.0190e−04	0.9667	**1.2413e−16**	0.9667	**1.2413e−16**	**0.9670**	5.7001e−04
	3	0.9744	5.2658e−04	0.9741	**0**	0.9743	1.2005e−04	**0.9752**	0.0016
	4	0.9761	6.7982e−04	0.9756	**3.7775e−04**	**0.9766**	5.3421e−04	0.9738	0.0018
	5	0.9788	0.0012	0.9796	**6.8327e−06**	0.9799	4.4271e−04	**0.9801**	0.0025
House	2	0.9185	0.0192	0.9265	**1.2413e−16**	0.9267	1.7441e−04	**0.9300**	0.0054
	3	0.9084	0.0247	0.9109	**0**	0.9108	5.2879e−04	**0.9370**	0.0056
	4	0.9064	0.0472	0.8869	**0.0058**	0.9129	0.0483	**0.9311**	0.0450
	5	0.9870	0.0040	0.9366	**0.0013**	0.9846	0.0021	**0.9877**	**0.0689**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 5
Comparison of algorithms (GSA, PSOGSA, GGSA, DS, BBO-DE, GAPSO and GSA-GA) taking the entropy as evaluation criteria in terms of the mean and standard deviation of CPU times (in seconds). The best results of each metric are shown in boldface.

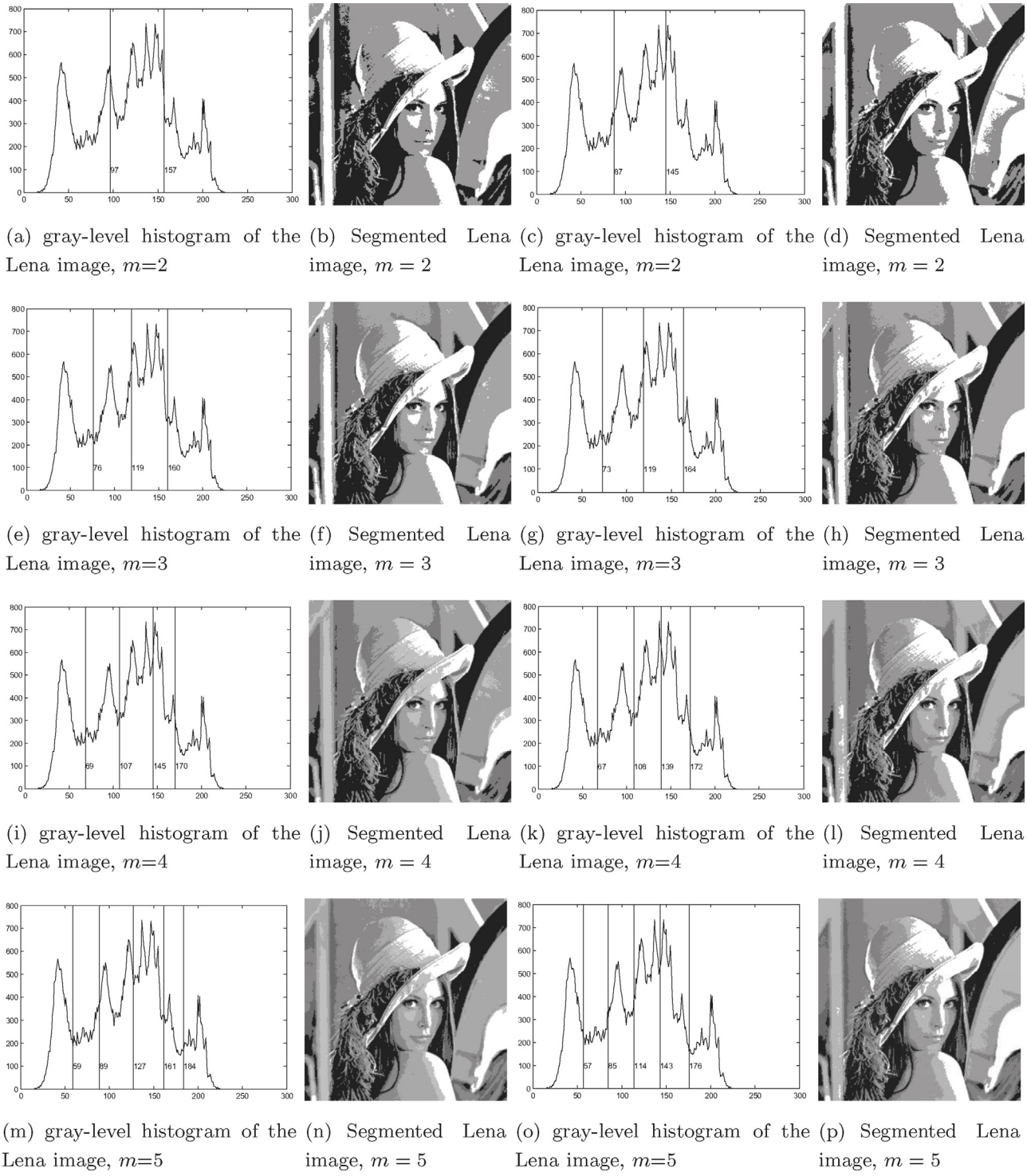
Image	m	GSA		PSOGSA		GGSA		DS		BBO-DE		GAPSO		GSA-GA	
		M_t	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*
Lena	2	1.0123	**0.0050**	1.6078	0.1544	1.6813	0.0985	1.1003	0.0335	3.0389	0.6785	1.0994	0.0472	**0.8662**	0.0247
	3	1.0846	0.0175	1.6504	0.5173	1.9292	0.3106	2.2027	0.3149	2.6781	0.1947	1.1186	0.0139	**0.9364**	**0.0103**
	4	1.8611	0.1690	1.1077	0.0292	1.6873	0.0493	1.9755	0.0518	2.7709	0.0963	1.1850	0.0312	**0.9336**	**0.0196**
	5	2.0468	0.2064	1.1798	0.0492	1.8288	0.2273	1.2759	0.0239	2.8885	0.0790	1.2234	0.0266	**1.0030**	**0.0136**
Cameraman	2	1.0777	0.0684	1.6021	0.1463	1.6269	0.0980	1.1102	0.0299	2.3975	0.0713	1.0718	0.0296	**0.8871**	**0.0269**
	3	1.0991	0.0316	2.2101	0.1230	1.8792	0.1820	1.1890	0.0333	2.6235	0.0952	1.1587	0.0374	**0.9122**	**0.0094**
	4	1.1883	0.0156	1.1615	0.0253	1.3987	0.0328	1.2408	0.0260	2.9404	0.0139	1.1943	0.0301	**0.9501**	**0.0155**
	5	1.1860	0.0343	1.1050	0.0365	2.1713	0.1982	1.2935	0.0292	3.0819	0.1056	1.2413	0.0342	**1.0093**	**0.0227**
Butterfly	2	1.6482	0.0488	1.5048	0.0453	2.7512	0.0971	1.6925	0.0526	3.5244	**0.0165**	1.6548	0.0359	**1.4509**	0.0486
	3	1.6827	0.0284	1.6592	0.0626	4.0021	0.2647	3.4636	0.1195	3.6308	0.0601	1.7165	0.0604	**1.4960**	**0.0237**
	4	1.6905	**0.0316**	1.6323	0.0317	2.9516	0.9764	2.9635	0.1818	3.8368	0.0990	1.8147	0.0466	**1.5057**	0.0499
	5	2.6506	0.8602	1.6747	0.0540	1.7674	0.0514	2.7210	0.6504	4.0203	0.0835	1.8058	0.0534	**1.5425**	**0.0357**
Landscape	2	3.0312	0.1716	1.5389	0.0568	1.6057	0.0402	1.6476	0.0589	3.4387	0.1852	1.5936	0.0582	**1.4272**	**0.0313**
	3	3.0218	0.2287	1.5456	**0.0280**	1.6533	0.0467	1.7508	0.0350	3.5607	0.0710	1.6741	0.0509	**1.4429**	0.0429
	4	2.9793	0.1455	1.5260	0.0729	1.7119	0.0289	1.7152	0.0390	3.7590	0.0570	1.7228	0.0544	**1.4256**	**0.0185**
	5	3.0528	0.2215	1.7579	0.1123	1.7272	0.0326	1.7954	0.0598	4.1630	0.1962	1.7902	0.0498	**1.5390**	**0.0617**
Starfish	2	1.6752	0.0931	1.5435	0.0454	1.8275	0.0396	1.6917	0.0475	3.5784	0.0723	1.6640	0.0480	**1.4481**	**0.0378**
	3	1.6696	0.0243	1.6303	0.0592	2.8540	0.0992	1.7757	0.0585	3.7421	**0.0082**	1.7456	0.0562	**1.4228**	0.0201
	4	1.6825	**0.0233**	1.6505	0.0308	3.7464	0.5911	1.8840	0.0591	3.8296	0.1134	1.7878	0.0520	**1.4758**	0.0320
	5	2.6214	0.9823	1.7040	0.0534	3.5613	0.3179	2.1350	0.5651	4.0223	0.1147	1.8236	0.0443	**1.5715**	**0.0414**
House	2	1.0488	0.0370	1.0765	0.0339	1.0323	**0.0126**	1.1322	0.0458	2.3825	0.1039	1.0707	0.0254	**0.9144**	0.0673
	3	1.1113	**0.0168**	1.0908	0.0334	1.1569	0.0519	1.1852	0.0508	2.5713	0.0492	1.1439	0.0259	**0.9246**	**0.0466**
	4	1.0932	0.0432	1.1090	0.0328	1.1449	0.0259	2.0981	0.5521	2.8502	0.2266	1.1865	0.0351	**0.9645**	**0.0249**
	5	1.2845	0.0360	1.1854	0.0347	1.1890	0.0270	1.9882	0.1042	2.9229	0.0846	1.2625	0.0288	**1.0268**	**0.0173**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 6
Comparison of algorithms (DS, BBO-DE, GAPSO and GSA-GA) taking the entropy as evaluation criteria in terms of best uniformity and corresponding thresholds. The best results of each metric are shown in boldface.

Image	m	DS		BBO-DE		GAPSO		GSA-GA	
		Th	u	*Th*	u	*Th*	u	*Th*	*u*
Lena	2	85,142	**0.9696**	81,150	0.9683	101,149	0.9676	97,157	0.9690
	3	72,129,161	0.9452	71,124,181	0.9749	69,108,173	0.9731	76,119,160	**0.9773**
	4	57,88,115,162	0.9750	81,99,153,182	0.9705	63,109,147,195	0.9764	69,107,145,170	**0.9809**
	5	61,83,115,139,170	0.9782	64,96,149,170,196	0.9697	80,104,120,150,188	0.9773	59,89,127,161,184	**0.9800**
Cameraman	2	118,199	0.9489	143,197	0.9077	98,195	**0.9637**	113,193	0.9546
	3	81,145,190	**0.9796**	99,150,189	0.9717	56,141,208	0.9783	48,113,199	0.9758
	4	68,112,163,214	0.9776	48,92,153,226	0.9794	65,102,143,208	0.9788	46,97,143,191	**0.9824**
	5	31,88,124,148,217	0.9829	22,78,143,167,217	0.9828	55,76,121,158,195	0.9831	40,82,115,146,194	**0.9836**
Butterfly	2	83,153	**0.9779**	93,146	0.9763	84,170	0.976	98,176	0.9738
	3	76,136,191	0.9761	65,134,176	0.9726	84,113,171	0.9745	74,117,171	**0.9769**
	4	91,130,169,231	0.9745	70,80,127,200	0.9706	57,106,156,204	0.9758	77,123,167,208	**0.9803**
	5	60,91,137,168,224	0.9793	74,114,163,203,248	0.9779	69,80,109,162,213	0.9768	52,84,131,174,219	**0.9808**
Landscape	2	94,172	0.9529	80,130	0.9524	82,143	**0.9541**	96,157	0.9499
	3	68,117,164	**0.9702**	68,118,153	0.9606	70,136,197	0.9643	74,133,198	0.9602
	4	59,84,132,182	**0.9851**	26,51,96,181	0.9706	25,94,135,190	0.9829	72,98,131,188	0.9621
	5	22,42,91,135,187	0.9843	37,56,101,170,187	0.9705	29,77,107,132,184	**0.9901**	48,93,124,169,195	0.9832
Starfish	2	97,176	0.9654	86,152	0.9673	89,157	**0.9674**	92,161	0.9671
	3	81,132,192	0.9724	54,130,178	0.9681	66,131,183	0.9730	68,120,178	**0.9753**
	4	64,109,157,209	0.9760	83,106,137,187	0.9707	37,88,127,169	0.9723	65,110,155,201	**0.9771**
	5	58,94,110,145,203	0.9781	82,107,169,194,215	0.9588	40,83,120,147,175	0.9762	43,80,114,154,198	**0.9804**
House	2	112,196	0.9351	68,99	0.9170	67,123	0.9446	111,194	**0.9354**
	3	63,110,200	0.9077	104,184,196	0.9785	109,152,196	**0.9830**	64,133,195	0.9630
	4	63,104,137,185	0.9828	51,96,163,205	0.9834	51,106,161,198	0.9834	52,82,145,197	**0.9786**
	5	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	**0.9887**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the Lena image, m=2 (b) Segmented Lena image, $m = 2$ (c) gray-level histogram of the Lena image, m=2 (d) Segmented Lena image, $m = 2$

(e) gray-level histogram of the Lena image, m=3 (f) Segmented Lena image, $m = 3$ (g) gray-level histogram of the Lena image, m=3 (h) Segmented Lena image, $m = 3$

(i) gray-level histogram of the Lena image, m=4 (j) Segmented Lena image, $m = 4$ (k) gray-level histogram of the Lena image, m=4 (l) Segmented Lena image, $m = 4$

(m) gray-level histogram of the Lena image, m=5 (n) Segmented Lena image, $m = 5$ (o) gray-level histogram of the Lena image, m=5 (p) Segmented Lena image, $m = 5$

Fig. 6. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the Lena image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

and the *Std_u* values of GSA-GA were slightly poor. This is due to the mutation operator makes particles explore more broadly and the experimental results may not stable enough in limited *FEs*. Furthermore, it is worthy to be emphasized that the mean of CPU times of GSA-GA are visible smaller than the comparison algorithms as shown in Table 5.

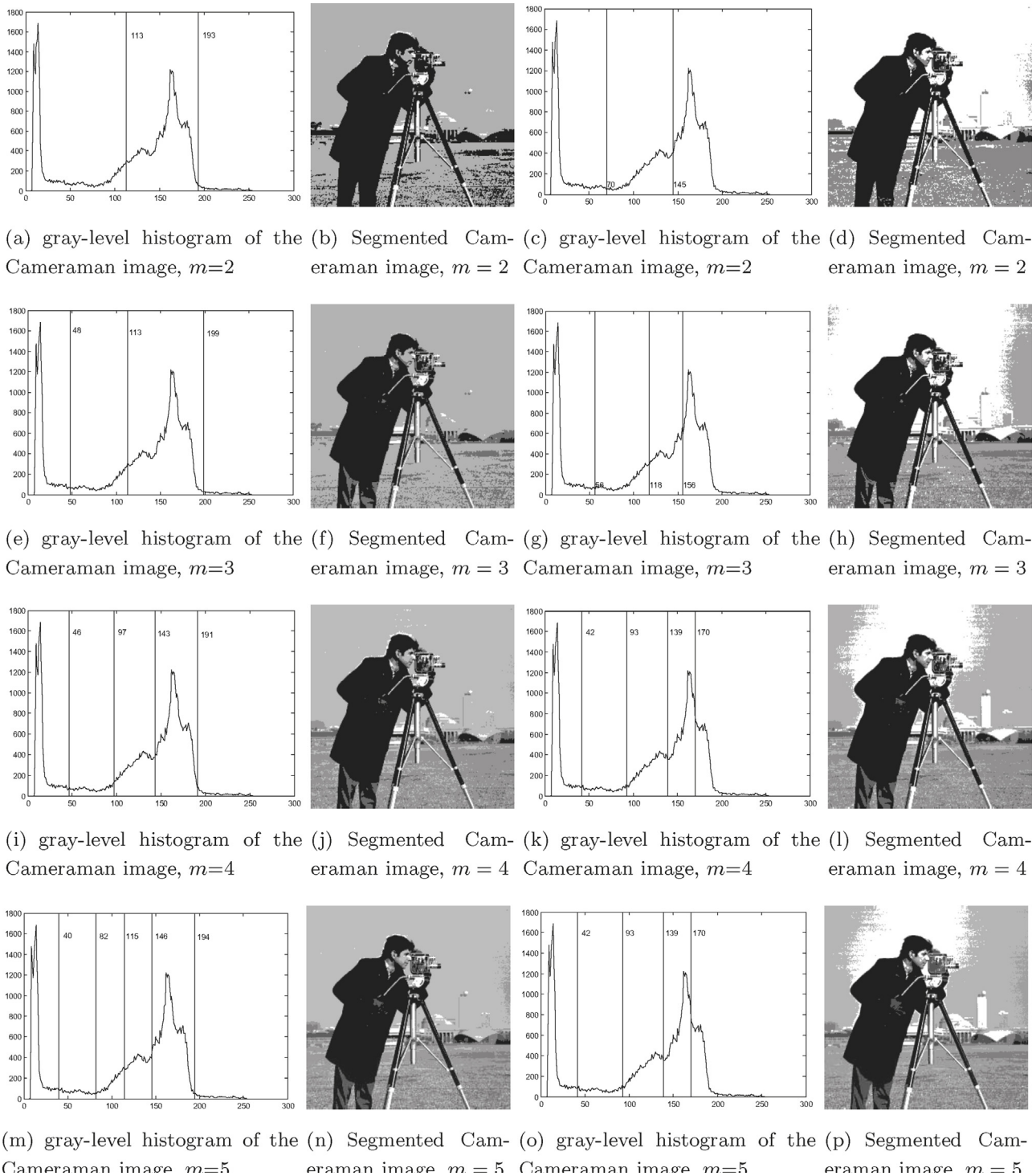
For a visual GSA-GA interpretation of the segmentation results, the segmented images and gray-level histograms labelled with thresholds were presented in Figs. 6–11 with m = 2–5. The first columns of the images show the histograms labelled with the thresholds found by the GSA-GA on the entropy, and the second columns give the segmented images through the threshold values obtained from the GSA-GA on the entropy fitness function. It was shown that the segmentation quality is better when m = 5 in each case, except for Lena. This visual finding was supported by the calculated uniformity for GSA-GA in Table 3, which showed a higher uniformity value as we increased the number of thresholds for a given image.

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 7
Comparison of algorithms (DS, BBO-DE, GAPSO and GSA-GA) taking the entropy as evaluation criteria in terms of the mean and standard deviation of uniformity. The best results of each metric are shown in boldface.

Image	m	DS		BBO-DE		GAPSO		GSA-GA	
		M_u	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*
Lena	2	0.9667	0.0027	0.9653	0.0043	0.9666	0.0013	**0.9687**	**9.0735e−04**
	3	0.9683	0.0064	0.9708	0.0058	0.9696	0.0027	**0.9771**	**7.5622e−04**
	4	0.9709	0.0041	0.9663	0.0060	0.9710	0.0067	**0.9791**	**0.0020**
	5	0.9687	0.0098	0.9651	0.0065	0.9700	0.0061	**0.9773**	**0.0034**
Cameraman	2	0.9186	0.0354	0.8811	0.0377	0.9225	0.0910	**0.9355**	**0.0135**
	3	**0.9728**	0.0080	0.9706	0.0035	0.9727	0.0058	0.9694	**0.0035**
	4	0.9687	0.0107	0.9791	**5.1999e−04**	0.9743	0.0110	**0.9814**	0.0026
	5	0.9776	0.0048	0.9806	**2.4610e−04**	0.9735	0.0128	**0.9816**	0.0037
Butterfly	2	0.9701	0.0035	**0.9752**	0.0026	0.9692	0.0053	0.9701	**0.0024**
	3	0.9723	0.0073	0.9669	0.0080	0.9680	0.0068	**0.9768**	**0.0068**
	4	0.9712	0.0033	0.9688	0.0025	0.9639	0.0104	**0.9785**	**7.2751e−04**
	5	0.9740	0.0040	0.9752	0.0038	0.9744	0.0029	**0.9805**	**0.0022**
Landscape	2	**0.9489**	0.0038	0.9498	0.0037	0.9469	0.0067	0.9459	**0.0037**
	3	**0.9573**	0.0108	0.948	0.0178	0.9464	0.0156	0.9550	**0.0090**
	4	0.9521	0.0322	0.9413	0.0414	**0.9646**	0.0196	0.9556	**0.0181**
	5	0.9803	0.0028	0.9699	**8.7058e−04**	0.9752	0.0095	**0.9808**	0.0065
Starfish	2	0.9634	0.0016	0.9665	0.0011	0.9566	0.0084	**0.9670**	**5.7001e−04**
	3	0.9695	0.0037	0.9669	0.0017	0.9687	0.0042	**0.9752**	**0.0016**
	4	0.9662	0.0101	0.9697	0.0025	0.9661	0.0061	**0.9738**	**0.0018**
	5	0.9595	0.0143	0.9473	0.0162	0.9693	0.0052	**0.9801**	**0.0025**
House	2	0.9109	0.0305	0.9103	0.0095	0.9162	0.0312	**0.9300**	**0.0054**
	3	0.8882	0.0241	0.9366	0.0062	0.9084	0.0517	**0.9370**	**0.0056**
	4	0.9306	0.0566	0.9218	**0.0023**	0.9288	0.0429	**0.9311**	0.0450
	5	0.9322	0.0583	0.9191	0.0916	0.9805	**0.0022**	**0.9877**	0.0689

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the Cameraman image, m=2 (b) Segmented Cameraman image, $m = 2$ (c) gray-level histogram of the Cameraman image, m=2 (d) Segmented Cameraman image, $m = 2$

(e) gray-level histogram of the Cameraman image, m=3 (f) Segmented Cameraman image, $m = 3$ (g) gray-level histogram of the Cameraman image, m=3 (h) Segmented Cameraman image, $m = 3$

(i) gray-level histogram of the Cameraman image, m=4 (j) Segmented Cameraman image, $m = 4$ (k) gray-level histogram of the Cameraman image, m=4 (l) Segmented Cameraman image, $m = 4$

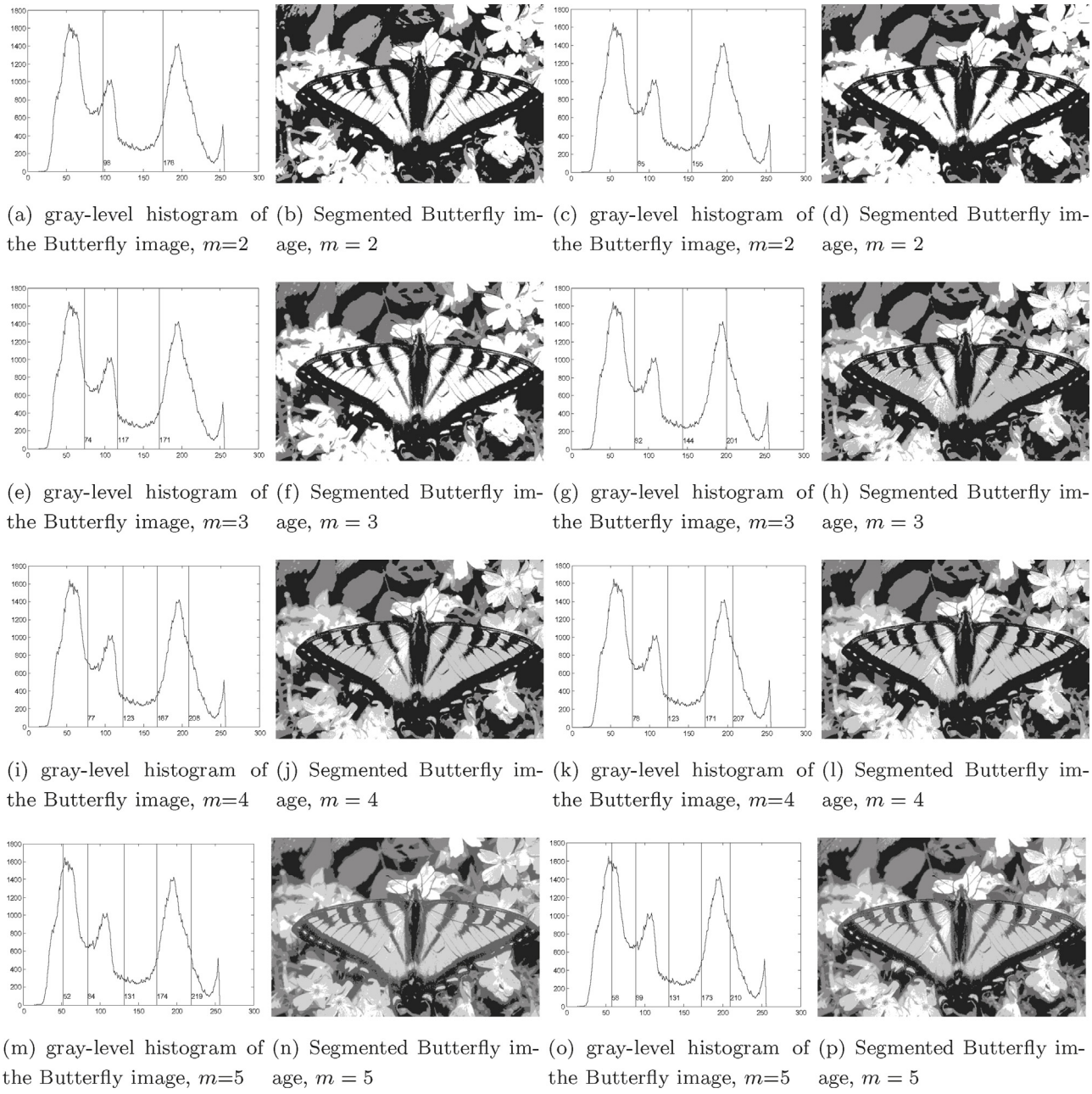
(m) gray-level histogram of the Cameraman image, m=5 (n) Segmented Cameraman image, $m = 5$ (o) gray-level histogram of the Cameraman image, m=5 (p) Segmented Cameraman image, $m = 5$

Fig. 7. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the Cameraman image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

To further demonstrate the superiority of GSA-GA, we carried out experiments on DS, BBO-DE, and GAPSO using the six benchmark images shown in Fig. 5. Table 6 presented the best uniformity (u) and the corresponding threshold values (Th) on all the test images produced by the algorithms after 30 runs. To test the stability of the algorithms, Table 7 presented the mean and stand deviation of uniformity (M_u and Std_u). Moreover, to test the computation complexity, the mean and stand deviation of CPU times (M_t and Std_t) after 30 runs were also reported in Table 5.

For images Lena, Cameraman, Butterfly, Starfish and House, when m = 2 and 3, the GSA-GA produced the best M_u and Std_u on almost all the test images as illustrated in Table 7. For m = 4 and 5, GSA-GA outperformed the other three algorithms completely in terms of the best uniformity and the M_u as shown in Tables 6 and 7. It thus can be inferred that the introducing of the GA's discrete mutation operator has enhanced the diversity of the population and alleviated the premature convergence problem. For the Landscape image, although GSA-GA has not performed the best on the

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the Butterfly image, m=2 (b) Segmented Butterfly image, $m = 2$ (c) gray-level histogram of the Butterfly image, m=2 (d) Segmented Butterfly image, $m = 2$

(e) gray-level histogram of the Butterfly image, m=3 (f) Segmented Butterfly image, $m = 3$ (g) gray-level histogram of the Butterfly image, m=3 (h) Segmented Butterfly image, $m = 3$

(i) gray-level histogram of the Butterfly image, m=4 (j) Segmented Butterfly image, $m = 4$ (k) gray-level histogram of the Butterfly image, m=4 (l) Segmented Butterfly image, $m = 4$

(m) gray-level histogram of the Butterfly image, m=5 (n) Segmented Butterfly image, $m = 5$ (o) gray-level histogram of the Butterfly image, m=5 (p) Segmented Butterfly image, $m = 5$

Fig. 8. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the Butterfly image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

best uniformity and *M_u*, it produced the best robustness for $m = 2$, 3, and 4 as illustrated by the *Std_u* shown in Table 7. The inferior robustness of DS, BBO-DE, and GAPSO may come from the influence of their control parameters, crossover and mutation operators as shown in Table 2. In addition, the CPU times shown in Table 5 also confirmed the low computational complexity of GSA-GA.

4.5. Experiment 2: Maximizing between-class variance

This section devotes to maximize the between-class variance criterion to construct an optimally segmented image for all of the seven algorithms. We first compared GSA-GA to the other three GSA variants. Experiments were performed on all the test images shown in Fig. 5. The best uniformity (u) and the corresponding threshold values (*Th*) produced by the algorithms were given in Table 8 after 30 runs. Similarly, to test the stability of the algorithms, Table 9 presented the mean and stand deviation of uniformity (*M_u* and *Std_u*). In addition, Table 10 provided the mean and stand deviation of CPU times (*M_t* and *Std_t*) after 30 runs.

As illustrated in Table 8, with the between-class variance criterion, GSA and GSA-GA yielded much better results than PSOGSA and GGSA. Moreover, although GSA can obtained comparative results with GSA-GA when thresholds are small, GSA-GA performed the best segmentation in most cases. This confirmed the effectiveness of the introduced GA operators. Meanwhile, GSA-GA yielded the best *M_u* in most cases as illustrated by Table 9. Although the mutation operator also caused a sightly poorer *Std_u* as shown in Table 9, GSA-GA consumed the least CPU times as shown in Table 10.

In Figs. 6–11, the third columns presented the histograms labelled with the thresholds produced by the GSA-GA on the between-class variance criterion, and the fourth columns displayed the segmentation images by the GSA-GA used the between-class

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 8
Comparison of algorithms (GSA, PSOGSA, GGSA, and GSA-GSA) taking the between-class variance as evaluation criteria in terms of best uniformity and corresponding thresholds. The best results of each metric are shown in boldface.

Image	m	GSA		PSOGSA		GGSA		GSA-GA	
		Th	u	*Th*	u	*Th*	u	*Th*	*u*
Lena	2	87,144	**0.9697**	87,145	**0.9697**	87,145	**0.9697**	87,145	**0.9697**
	3	73,119,164	0.9776	74,120,164	**0.9777**	74,120,164	**0.9777**	73,119,164	0.9776
	4	65,103,137,173	0.9817	85,90,149,157	0.9530	66,99,163,177	0.9664	67,108,139,172	**0.9820**
	5	57,85,113,142,176	**0.9834**	73,86,119,157,165	0.9714	71,78,119,163,176	0.9722	57,85,114,143,176	**0.9834**
Cameraman	2	70,144	**0.9844**	70,144	**0.9844**	70,144	**0.9844**	70,145	**0.9844**
	3	58,119,156	**0.9840**	71,89,147	0.9792	59,119,156	**0.9840**	56,118,156	**0.9840**
	4	44,94,133,169	0.9853	69,70,144,165	0.9765	42,94,140,171	**0.9860**	42,93,139,170	**0.9860**
	5	41,90,128,159,200	0.9861	58,68,86,131,164	0.9795	29,78,118,147,173	0.9873	29,75,119,148,173	**0.9874**
Butterfly	2	85,155	**0.9780**	85,156	**0.9780**	85,155	**0.9780**	85,155	**0.9780**
	3	83,145,201	**0.9789**	93,135,187	0.9758	82,145,201	**0.9789**	82,144,201	**0.9789**
	4	76,120,168,208	**0.9803**	88,134,169,254	0.9681	77,119,175,254	0.9705	78,123,171,207	**0.9803**
	5	60,89,127,167,210	0.9823	82,145,165,206,254	0.9716	40,83,117,149,182	0.9710	58,89,131,173,210	**0.9824**
Landscape	2	54,144	**0.9635**	54,144	**0.9635**	54,144	**0.9635**	53,144	**0.9635**
	3	50,111,174	**0.9815**	49,110,174	**0.9815**	47,110,174	**0.9815**	49,110,176	**0.9815**
	4	91,132,182	**0.9868**	45,92,132,182	**0.9868**	45,92,132,182	**0.9868**	49,96,136,181	**0.9868**
	5	38,80,108,139,185	0.9916	36,51,95,146,200	0.9773	41,80,109,139,177	0.9916	40,81,106,137,182	**0.9917**
Starfish	2	85,157	**0.9676**	85,157	**0.9676**	85,157	**0.9676**	85,157	**0.9676**
	3	67,119,179	0.9753	69,120,178	0.9754	69,120,178	**0.9754**	68,119,178	**0.9754**
	4	59,100,140,187	0.9788	79,85,157,174	0.9516	60,101,138,187	**0.9789**	60,102,139,186	0.9788
	5	48,86,119,146,195	0.9809	69,79,120,126,178	0.9673	42,69,119,175,184	0.9697	51,84,117,150,194	0.9816
House	2	96,155	**0.9875**	96,155	**0.9875**	96,155	**0.9875**	96,155	0.9875
	3	79,110,154	0.9862	90,109,147	0.9847	81,112,158	**0.9864**	80,111,158	**0.9864**
	4	83,113,155,202	**0.9876**	96,109,155,197	0.9851	78,93,109,149	0.9823	81,109,153,201	0.9875
	5	80,102,115,154,194	0.9812	96,108,152,199,204	0.9815	80,110,148,164,238	0.9814	76,102,132,167,199	**0.9888**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 9
Comparison of algorithms (GSA, PSOGSA, GGSA, and GSA-GSA) taking the between-class variance as evaluation criteria in terms of the mean and standard deviation of uniformity. The best results of each metric are shown in boldface.

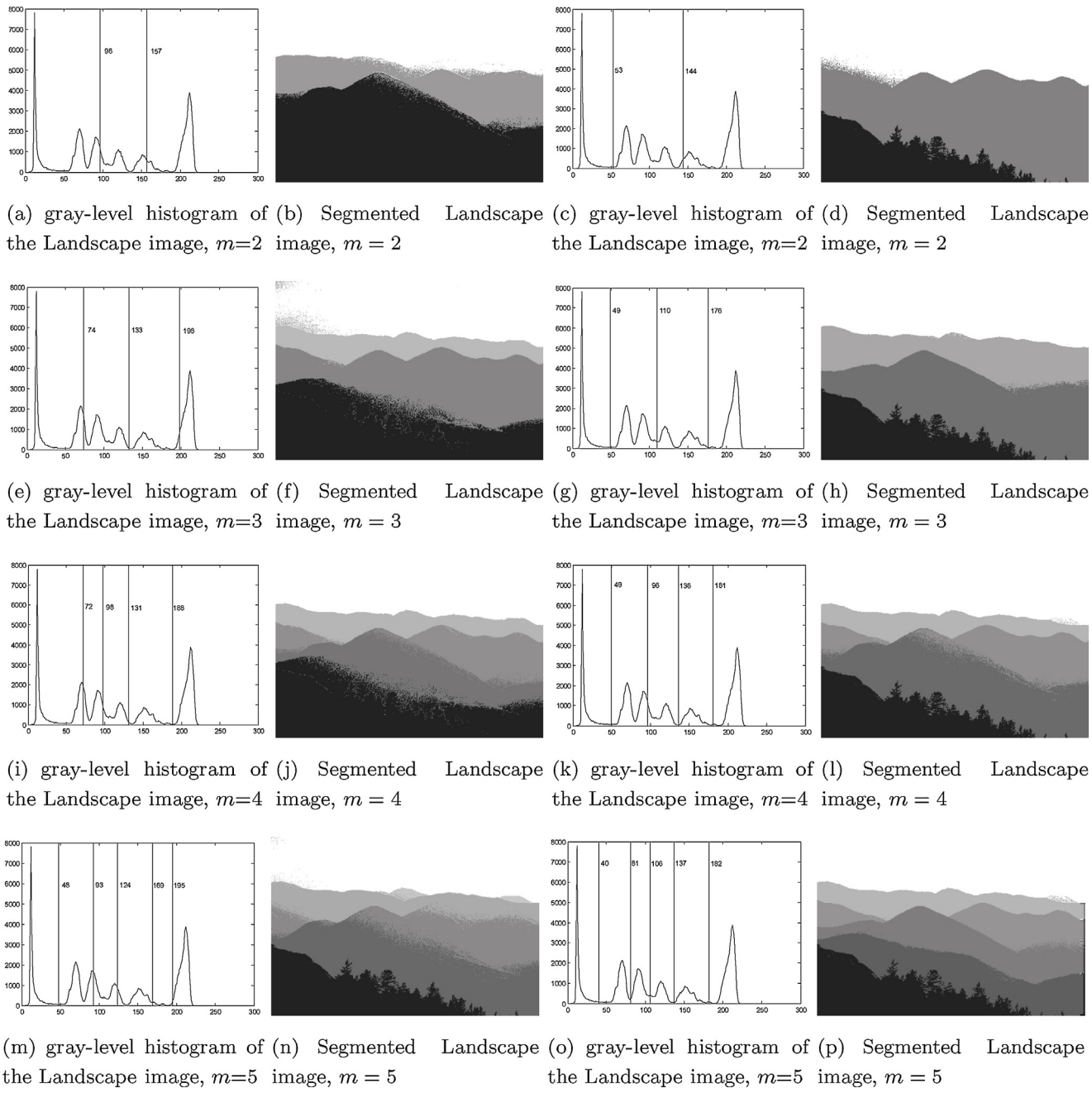
Image	*m*	GSA		PSOGSA		GGSA		GSA-GA	
		M_u	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*
Lena	2	**0.9697**	1.0912e−05	**0.9697**	**1.2413e−16**	**0.9697**	**1.2413e−16**	**0.9697**	1.0660e−05
	3	**0.9776**	9.0312e−05	0.9618	0.0219	0.9604	0.0246	**0.9776**	**2.2505e−05**
	4	0.9815	2.7102e−04	0.9450	**0.0058**	0.9509	0.0119	**0.9818**	0.0295
	5	0.9780	**0.0070**	0.9315	0.0408	0.9678	0.0074	**0.9801**	0.0347
Cameraman	2	0.9844	**1.2413e−16**	0.9844	**1.2413e−16**	0.9844	**1.2413e−16**	**0.9844**	2.8192e−05
	3	0.9829	0.0023	0.9715	0.0045	0.9788	0.0036	**0.9833**	**6.7899e−04**
	4	0.9848	**4.9043e−05**	0.9636	0.0155	0.9751	0.0076	**0.9859**	5.3867e−04
	5	0.9711	**0.0022**	0.9767	**0.0022**	0.9684	0.0191	**0.9858**	0.0310
Butterfly	2	**0.9780**	4.3573e−06	**0.9780**	**0**	**0.9780**	4.3573e−06	**0.9780**	**4.3573e−06**
	3	0.9785	**7.1317e−05**	0.9534	0.0146	0.9622	0.0187	**0.9789**	4.7317e−04
	4	**0.9775**	0.0060	0.9444	0.0218	0.9644	**0.0056**	0.9733	0.0117
	5	0.9657	0.0134	0.9494	0.0230	0.9669	**0.0045**	**0.9762**	0.0273
Landscape	2	**0.9635**	**1.2413e−16**	**0.9635**	**1.2413e−16**	**0.9635**	**1.2413e−16**	**0.9635**	4.1819e−05
	3	**0.9815**	1.4871e−05	**0.9815**	**0**	**0.9715**	0.0224	**0.9815**	6.5952e−05
	4	**0.9868**	**2.8765e−05**	0.9641	0.0219	0.9538	0.0252	0.9839	0.0038
	5	0.9864	**0.0043**	0.9299	0.0532	0.9719	0.0134	**0.9894**	**0.0043**
Starfish	2	**0.9676**	3.3622e−05	**0.9676**	**0**	**0.9676**	**0**	**0.9676**	1.2819e−05
	3	0.9752	2.2472e−04	**0.9754**	**1.2413e−16**	0.9753	7.2837e−06	0.9752	1.3188e−04
	4	**0.9709**	0.0140	0.9443	**0.0046**	0.9619	0.0164	0.9511	0.0609
	5	0.9750	0.0083	0.9644	**0.0037**	0.9651	0.0055	**0.9778**	0.0058
House	2	**0.9875**	**1.2413e−16**	**0.9875**	**1.2413e−16**	**0.9875**	**1.2413e−16**	**0.9875**	1.0286e−05
	3	0.9848	**1.7003e−04**	0.9758	0.0157	0.9837	0.0025	**0.9863**	0.0020
	4	**0.9866**	**0.0018**	0.9695	0.0245	0.9791	0.0019	0.9811	0.0071
	5	0.9812	0.0068	0.9757	0.0044	0.9802	**0.0012**	**0.9853**	0.0055

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 10
Comparison of algorithms (GSA, PSOGSA, GGSA, DS, BBO-DE, GAPSO and GSA-GSA) taking the between-class variance as evaluation criteria in terms of the mean and standard deviation of CPU times (in seconds). The best results of each metric are shown in boldface.

Image	m	GSA		PSOGSA		GGSA		DS		BBO-DE		GAPSO		GSA-GA	
		M_t	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*	*M_t*	*Std_t*
Lena	2	1.0059	0.0130	0.9534	0.0441	0.9573	0.0430	2.0733	0.2590	2.3008	0.1345	0.9848	**0.0080**	**0.4285**	0.0195
	3	0.8708	**0.0041**	0.9799	0.0357	0.9528	0.0397	1.2432	0.1978	2.3294	0.0359	0.9907	0.0323	**0.6108**	0.0198
	4	1.0073	**0.0157**	1.0411	0.0553	0.9718	0.0253	1.0501	0.0360	2.4710	0.0585	1.0708	0.0692	**0.6172**	0.0173
	5	0.9902	**0.0179**	1.0450	0.0599	1.0577	0.0339	1.0775	0.0189	2.6076	0.0402	1.0892	0.0290	**0.6288**	0.0182
Cameraman	2	0.9125	0.0401	0.9518	0.0447	0.9348	0.0308	1.0473	0.0383	2.2599	0.0386	1.0404	0.0286	**0.6440**	**0.0158**
	3	0.9419	0.0323	1.0877	0.0721	0.9565	0.0452	1.1045	0.0384	2.4211	0.1275	1.0582	**0.0054**	**0.6632**	0.0212
	4	0.9831	0.0505	1.0220	**0.0262**	0.9755	0.0466	1.1165	0.5680	2.4634	0.0731	1.0623	0.0320	**0.7597**	0.3000
	5	0.9957	0.0342	1.0511	0.0450	0.9731	**0.0108**	1.1282	0.0403	2.6330	0.0914	1.0773	0.0192	**0.7989**	0.0303
Butterfly	2	1.6591	0.0806	1.5075	**0.0395**	1.5489	0.0988	1.6806	0.0698	3.2958	0.0682	1.6232	0.0539	**1.0284**	0.0414
	3	3.0343	0.3297	1.5417	**0.0542**	1.5705	0.0577	1.7599	0.0435	3.4172	0.0916	1.6035	0.0679	**1.4168**	0.4246
	4	3.1751	0.3484	1.5702	0.0482	1.5600	0.0512	1.6966	0.0537	3.6368	0.0574	1.6414	**0.0358**	**1.0178**	0.0728
	5	1.6562	0.3875	1.6358	0.1053	1.6872	0.0355	1.7234	0.0764	3.6506	0.1005	1.6540	0.0638	**1.0307**	**0.0485**
Landscape	2	1.7243	**0.0079**	1.5505	0.0574	1.5184	0.1049	1.5976	0.0308	3.2374	0.1537	1.5981	0.0471	**1.1415**	0.0303
	3	1.8753	**0.0526**	1.5703	0.0661	1.5202	0.0723	1.7200	0.1082	3.3970	0.0469	1.6406	0.0701	**1.2450**	0.0678
	4	1.8203	**0.0294**	1.6220	0.1319	1.5196	0.0582	1.6372	0.0602	3.5771	0.1267	1.6272	0.0607	**1.2379**	0.0585
	5	1.8411	**0.0337**	1.6379	0.0530	1.7089	0.0970	1.6594	0.0659	3.6800	0.1233	1.5664	0.0398	**1.3121**	0.0380
Starfish	2	1.8541	**0.0160**	1.6135	0.0658	1.5807	0.0689	2.0253	0.1193	3.3836	0.0596	1.6734	0.0499	**1.1823**	0.1074
	3	1.7910	**0.0226**	1.6069	0.0743	1.5804	0.0551	2.0577	0.1346	3.5209	0.0891	1.6804	0.0453	**1.2393**	0.1189
	4	1.8715	0.0487	1.6096	0.0493	1.5922	0.0691	2.1190	0.1291	3.7044	0.0760	1.6828	0.0378	**1.2682**	**0.0247**
	5	1.8072	0.0887	1.6371	**0.0330**	2.3991	0.0902	2.0510	0.4385	3.7020	0.0998	1.7633	0.0667	**1.2777**	0.0487
House	2	1.1504	0.1510	0.9398	0.0329	0.8722	0.0267	1.0113	0.0338	2.2981	0.0551	0.9848	**0.0214**	**0.7026**	0.0594
	3	0.9435	0.0243	0.9785	0.0558	0.8744	**0.0140**	0.9876	0.0211	2.3724	0.0787	1.0608	0.0528	**0.7657**	0.0173
	4	1.0936	0.0347	1.0273	0.0499	1.1108	0.0875	1.1089	0.0312	2.5155	0.0887	1.0065	**0.0035**	**0.7762**	0.0269
	5	1.0838	0.0448	1.0335	0.0430	1.0464	0.0222	1.2391	0.0991	2.6006	0.0323	1.0453	**0.0290**	**0.7325**	0.0326

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the Landscape image, m=2 (b) Segmented Landscape image, $m = 2$ (c) gray-level histogram of the Landscape image, m=2 (d) Segmented Landscape image, $m = 2$

(e) gray-level histogram of the Landscape image, m=3 (f) Segmented Landscape image, $m = 3$ (g) gray-level histogram of the Landscape image, m=3 (h) Segmented Landscape image, $m = 3$

(i) gray-level histogram of the Landscape image, m=4 (j) Segmented Landscape image, $m = 4$ (k) gray-level histogram of the Landscape image, m=4 (l) Segmented Landscape image, $m = 4$

(m) gray-level histogram of the Landscape image, m=5 (n) Segmented Landscape image, $m = 5$ (o) gray-level histogram of the Landscape image, m=5 (p) Segmented Landscape image, $m = 5$

Fig. 9. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the Landscape image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

variance criterion. As indicated by Figs. 6–11, the optimal threshold values produced by the GSA-GA with the between-class variance criterion performed more suitable segmentation. Take the Lena image as example: when $m = 2$, Fig. 6(c) gave the thresholds labelled at the valley with 87 and 145, while Fig. 6(a) showed the thresholds labelled at the valley with 97 and 157. This arises from the reason that the between-class variance criterion fully considered the separability between groups. This visual finding was also supported by the performance metric uniformity of GSA-GA in Tables 3 and 8.

To further investigate the superiority of GSA-GA, we compared GSA-GSA with the three new and high quality meta-heuristic algorithms DS, BBO-DE, and GAPSO using the between-class variance. Experiments were also performed on all the test images shown in Fig. 5. The best uniformity (u) and the corresponding threshold values (Th) produced by the algorithms were given in Table 11 after 30 runs. Afterwards, the mean and stand deviation of uniformity (M_u and Std_u) of 30 runs is exhibited in Table 12. In addition, the corresponding mean and stand deviation of CPU times (M_t and Std_t) are provided in Table 10.

As illustrated in Table 11, with the between-class variance criterion, GSA-GA yielded the best results in all the test images on $m = 2$, 3, and 4. Moreover, for $m = 2$, 3, and 4, GSA-GA outperformed the other three algorithms in almost all the cases with respect to the M_u and Std_u as shown in Table 12. When $m = 5$, although the best uniformity is slightly worse than DS, it yielded the best M_u as shown in Table 12. Besides, the mean CPU times in Table tab:F1-F6 Speed and Reliability between-class reveal that the computational complexity of GSA-GSA is apparently lower than those comparison algorithms.

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 11
Comparison of algorithms (DS, BBO-DE, GAPSO and GSA-GSA) taking the between-class variance as evaluation criteria in terms of best uniformity and corresponding thresholds. The best results of each metric are shown in boldface.

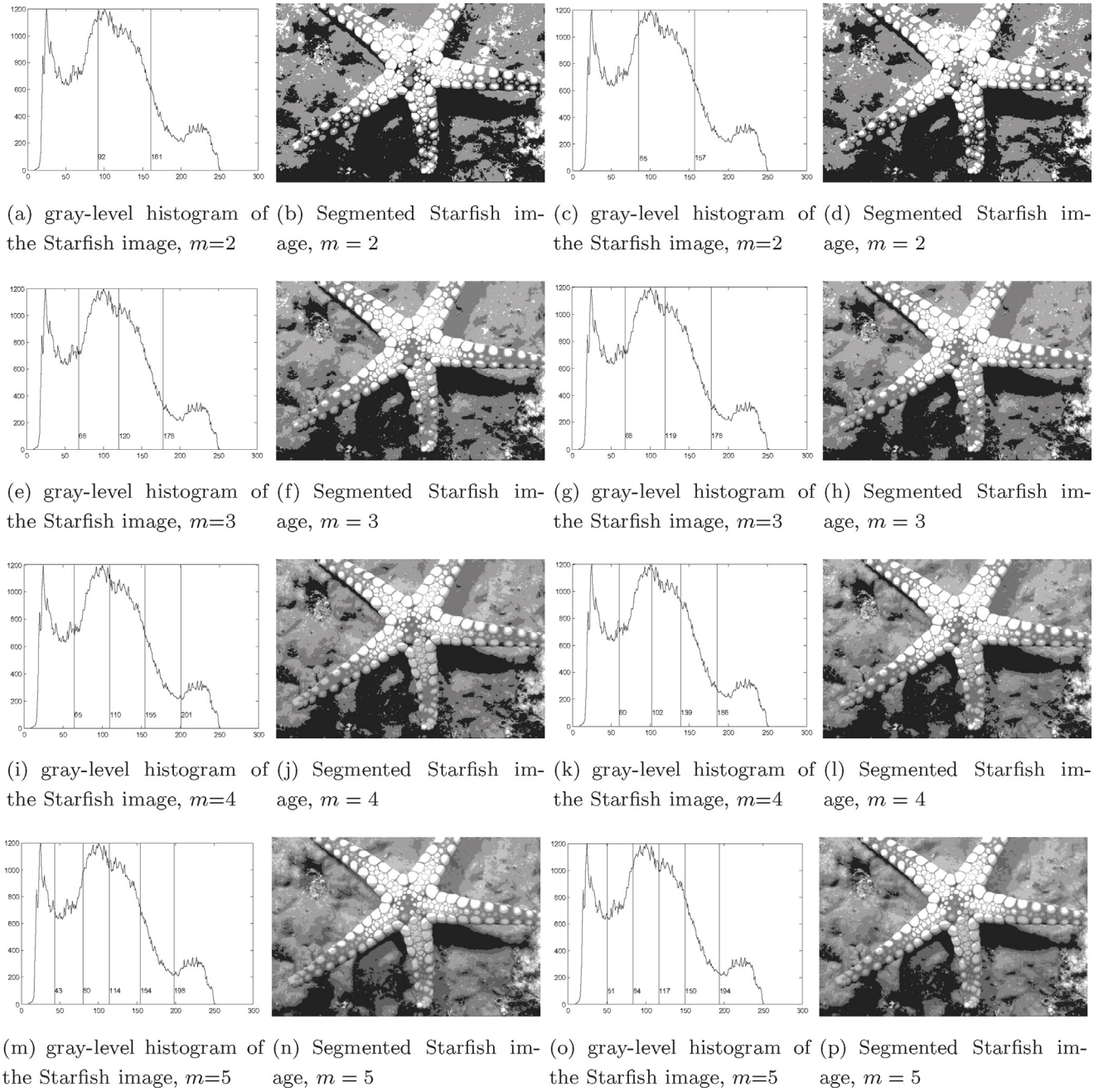
Image	m	DS		BBO-DE		GAPSO		GSA-GA	
		Th	u	*Th*	u	*Th*	u	*Th*	u
Lena	2	91,144	0.9693	79,144	0.969	95,154	0.9692	87,145	**0.9697**
	3	69,111,151	0.9744	79,109,160	0.975	66,118,172	0.9758	73,119,164	**0.9776**
	4	64,103,138,182	0.9808	74,114,144,167	0.980	62,110,127,160	0.9768	67,108,139,172	**0.9820**
	5	74,110,148,166,188	0.9786	56,85,110,133,166	0.981	65,112,142,172,202	0.9788	57,85,114,143,176	**0.9834**
Cameraman	2	73,150	0.9840	69,141	0.984	78,148	0.9840	70,145	**0.9844**
	3	72,115,149	0.9822	66,113,160	0.982	34,89,155	0.9818	56,118,156	**0.9840**
	4	41,66,121,154	0.9829	46,91,123,160	0.984	49,87,122,159	0.9838	42,93,139,170	**0.9860**
	5	41,86,123,150,172	**0.9875**	32,66,116,153,173	0.986	46,76,126,159,201	0.9848	29,75,119,148,173	0.9874
Butterfly	2	84,156	**0.9780**	89,151	0.978	88,150	0.9776	85,155	**0.9780**
	3	88,150,216	0.9772	88,166,224	0.974	771,334,202	0.9775	82,144,201	**0.9789**
	4	74,138,167,202	0.9773	65,87,160,212	0.976	84,129,180,228	0.9778	78,123,171,207	**0.9803**
	5	50,69,107,161,206	0.9774	67,115,130,179,202	0.974	66,86,137,189,211	0.9792	58,89,131,173,210	**0.9824**
Landscape	2	48,145	0.9634	50,146	0.963	44,141	0.9631	53,144	**0.9635**
	3	29,101,180	0.9784	55,107,186	0.981	31,112,170	0.9798	49,110,176	**0.9815**
	4	41,103,137,183	0.9860	60,77,133,163	0.979	58,104,138,180	0.9856	49,96,136,181	**0.9868**
	5	44,86,103,145,180	0.9902	58,91,109,132,187	0.988	51,74,110,146,195	0.9887	40,81,106,137,182	**0.9917**
Starfish	2	78,153	0.9671	82,148	0.967	85,155	**0.9676**	85,157	**0.9676**
	3	70,127,187	0.9744	65,109,189	0.972	68,106,171	0.9728	68,119,178	**0.9754**
	4	66,113,159,213	0.9752	66,104,142,209	0.975	50,95,127,171	0.9759	60,102,139,186	**0.9788**
	5	47,70,105,144,190	0.9799	60,89,132,157,204	0.979	55,73,104,142,184	0.9789	51,84,117,150,194	**0.9816**
House	2	103,157	0.9871	91,145	0.987	99,156	0.9874	96,155	**0.9875**
	3	81,109,156	0.9863	81,106,165	0.986	74,110,169	0.9853	80,111,158	**0.9864**
	4	76,106,162,204	0.9869	85,109,133,164	0.986	83,125,160,216	0.9850	81,109,153,201	**0.9875**
	5	78,91,127,175,213	0.9848	66,104,137,177,199	0.988	90,115,152,184,212	0.9870	76,102,132,167,199	**0.9888**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 12
Comparison of algorithms (DS, BBO-DE, GAPSO and GSA-GSA) taking the between-class variance as evaluation criteria in terms of the mean and standard deviation of uniformity. The best results of each metric are shown in boldface.

Image	*m*	DS		BBO-DE		GAPSO		GSA-GA	
		M_u	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*	*M_u*	*Std_u*
Lena	2	0.9685	0.0008	0.9677	0.0021	0.9652	0.0029	**0.9697**	**1.0660e−05**
	3	0.9652	0.0070	0.9683	0.0062	0.9723	0.0029	**0.9776**	**2.2505e−05**
	4	0.9735	0.0084	0.9740	0.0055	0.9743	**0.0028**	**0.9818**	0.0295
	5	0.9772	**0.0019**	0.9746	0.0060	0.9767	0.0022	**0.9801**	0.0347
Cameraman	2	0.9823	0.0012	0.9837	6.4812e−04	0.9809	0.0044	**0.9844**	**2.8192e−05**
	3	0.9799	0.0021	0.9800	0.0027	0.9801	0.0016	**0.9833**	**6.7899e−04**
	4	0.9802	0.0019	0.9820	0.0015	0.9808	0.0021	**0.9859**	**5.3867e−04**
	5	0.9821	0.0042	0.9828	**0.0021**	0.9821	0.0030	**0.9858**	0.0310
Butterfly	2	0.9752	0.0022	0.9759	0.0017	0.9767	0.0011	**0.9780**	**4.3573e−06**
	3	0.9747	0.0026	0.9716	0.0019	0.9735	0.0038	**0.9789**	**4.7317e−04**
	4	0.9744	0.0028	0.9729	**0.0027**	**0.9760**	**0.0027**	0.9733	0.0117
	5	0.9749	**0.0025**	0.9712	0.0030	0.9753	0.0029	**0.9762**	0.0273
Landscape	2	0.9626	0.0015	0.9622	0.0020	0.9610	0.0022	**0.9635**	**4.1819e−05**
	3	0.9757	0.0016	0.9738	0.0087	0.9715	0.0084	**0.9815**	**6.5952e−05**
	4	0.9768	0.0068	0.9765	**0.0020**	0.9772	0.0073	**0.9839**	0.0038
	5	0.9842	0.0034	0.9815	0.0049	0.9834	0.0039	**0.9894**	**0.0043**
Starfish	2	0.9657	0.0017	0.9650	0.0033	0.9661	0.0019	**0.9676**	**1.2819e−05**
	3	0.9681	0.0053	0.9634	0.0072	0.9699	0.0022	**0.9752**	**1.3188e−04**
	4	0.9695	0.0038	**0.9719**	0.0025	0.9715	0.0058	0.9511	0.0609
	5	0.9739	0.0105	0.9706	0.0062	0.9736	0.0104	**0.9778**	**0.0058**
House	2	0.9830	0.0059	0.9864	2.8152e−04	0.9867	6.1073e−04	**0.9875**	**1.0286e−05**
	3	0.9849	0.0015	0.9843	0.0011	0.9847	**7.0360e−04**	**0.9863**	0.0020
	4	0.9840	0.0022	0.9837	0.0030	0.9839	**8.6663e−04**	0.9811	0.0071
	5	0.9834	0.0066	0.9859	0.0058	0.9833	0.0065	**0.9853**	**0.0055**

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the Starfish image, m=2 (b) Segmented Starfish image, $m = 2$ (c) gray-level histogram of the Starfish image, m=2 (d) Segmented Starfish image, $m = 2$

(e) gray-level histogram of the Starfish image, m=3 (f) Segmented Starfish image, $m = 3$ (g) gray-level histogram of the Starfish image, m=3 (h) Segmented Starfish image, $m = 3$

(i) gray-level histogram of the Starfish image, m=4 (j) Segmented Starfish image, $m = 4$ (k) gray-level histogram of the Starfish image, m=4 (l) Segmented Starfish image, $m = 4$

(m) gray-level histogram of the Starfish image, m=5 (n) Segmented Starfish image, $m = 5$ (o) gray-level histogram of the Starfish image, m=5 (p) Segmented Starfish image, $m = 5$

Fig. 10. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the Starfish image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

4.6. *Running time analysis using Student's* t*-test*

To statistically analyze the computational complexity in Tables 5 and 10, a parametric significance proof known as the Student's t-test was conducted in this section [59]. This test allows assessing result differences among two related methods (one of the methods is chosen as control method). In this paper, the GSA-GA algorithm was chosen as the control algorithm and was compared with other six comparison algorithms in terms of the mean CPU times. A null hypothesis is assumed 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 $\alpha = 0.05$. If in any test a t-value that is smaller than or equal to the critical value '−2.045' is produced, the null hypothesis for that test is rejected [60,61].

Tables 13 and 14 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 entropy and between-class variance, respectively) versus (VS) other comparison algorithms. As illustrated in Table 13, for all the test images, the produced t-values for the experiments on the entropy criterion were smaller than the critical value. The results indicated that GSA-GA performed significantly fast image segmentation compared with the other six algorithms. Similarly, as shown in Table 14, for test images Lena, Cameraman, Landscape, Starfish and House, the produced t-values for the experiments on the between-class variance criterion are smaller than the critical value. These results suggested that GSA-GA spend significant short running time to achieve multi-level thresholding compared with the other six comparison algorithms on these test images. For the test image Butterfly, GSA-GA

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 13
Statistical analysis of 30 runs for the experiments on the entropy criterion. *t*-Value: probability value of the *t*-test, $h = 0$ indicates null hypothesis cannot be rejected at the 0.05 significance level, $h = 1$ indicates null hypothesis can be rejected.

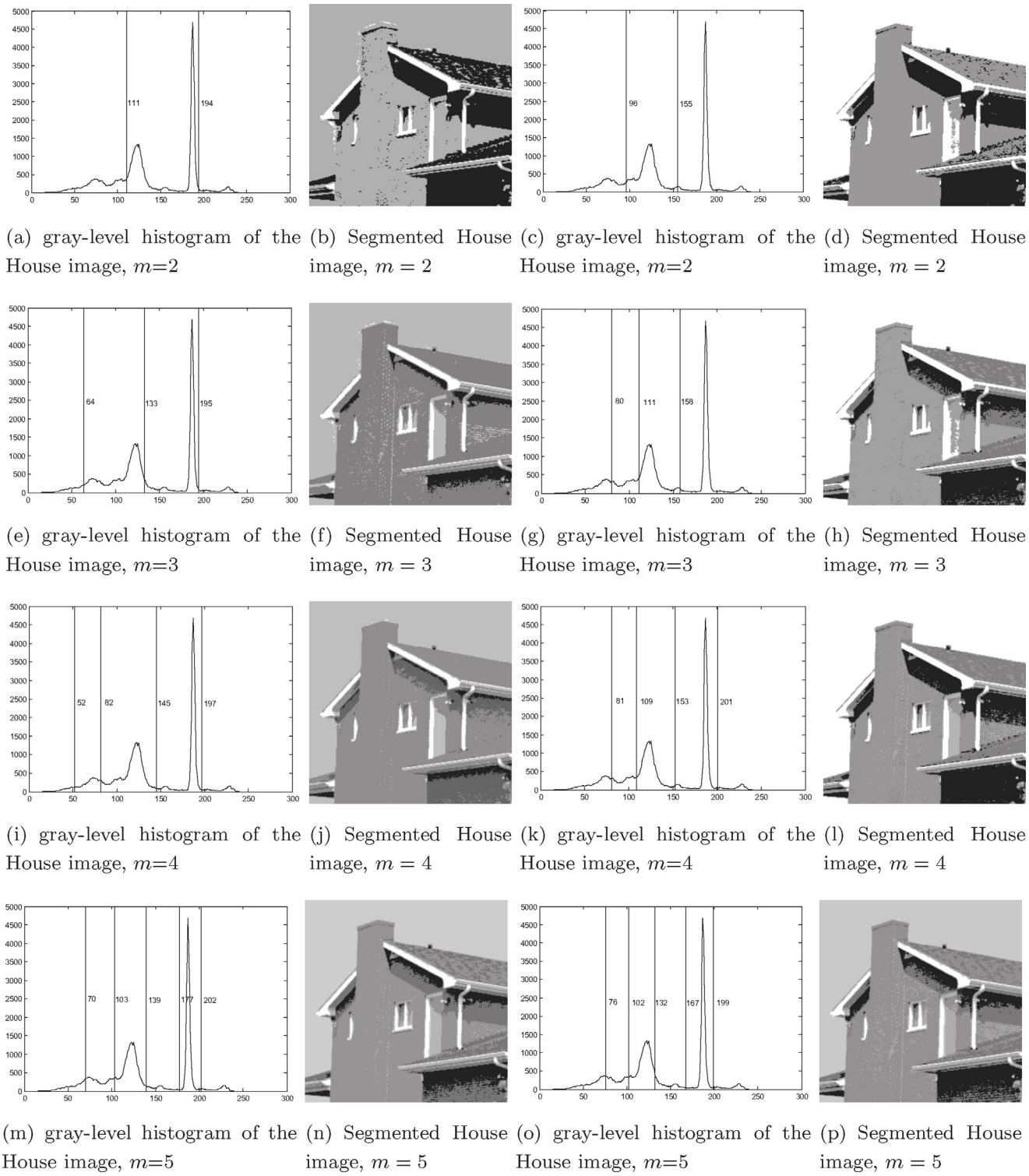
Image	*m*	GSA-GA VS GSA		GSA-GA VS PSOGSA		GSA-GA VS GGSA		GSA-GA VS DS		GSA-GA VS BBO-DE		GSA-GA VS GAPSO	
		t-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*
Lena	2	−31.7536	1	−25.9774	1	−43.9636	1	−17.5276	1	−23.9766	1	−30.8067	1
	3	−39.9743	1	−7.5584	1	−17.4978	1	−48.9284	1	−57.6840	1	−22.0137	1
	4	−29.8598	1	−27.1150	1	−77.8121	1	−102.4001	1	−37.3715	1	−103.0390	1
	5	−27.6393	1	−18.9709	1	−19.8637	1	−128.8303	1	−40.4076	1	−54.3569	1
Cameraman	2	−14.2036	1	−26.3271	1	−39.8726	1	−108.5589	1	−25.2929	1	−30.3824	1
	3	−31.0507	1	−57.6278	1	−29.0628	1	−97.9813	1	−35.0110	1	−43.8162	1
	4	−59.3273	1	−39.0248	1	−67.7294	1	−523.6061	1	−39.5062	1	−52.6016	1
	5	−23.5302	1	−12.1948	1	−31.9031	1	−105.1001	1	−30.9569	1	−42.0875	1
Butterfly	2	−15.6907	1	−4.4436	1	−65.5904	1	−221.2786	1	−18.4836	1	−18.4779	1
	3	−27.6454	1	−13.3543	1	−51.6501	1	−180.9911	1	−18.6138	1	−88.4611	1
	4	−17.1372	1	−11.7294	1	−8.1004	1	−115.1669	1	−24.7887	1	−42.3538	1
	5	−7.0496	1	−11.1856	1	−19.6836	1	−149.4465	1	−22.4514	1	−9.9096	1
Landscape	2	−50.3664	1	−9.4337	1	−19.1897	1	−58.6576	1	−13.7920	1	−18.0986	1
	3	−37.1655	1	−10.9803	1	−18.1729	1	−139.8321	1	−19.0233	1	−30.4597	1
	4	−58.0206	1	−7.3116	1	−45.6992	1	−213.2686	1	−28.3300	1	−36.7471	1
	5	−36.0602	1	−9.3572	1	−14.7717	1	−69.8791	1	−17.3525	1	−16.3442	1
Starfish	2	−12.3792	1	−8.8450	1	−37.9590	1	−143.0179	1	−19.3550	1	−21.9793	1
	3	−42.8651	1	−18.1788	1	−77.4484	1	−585.1832	1	−29.6224	1	−31.2482	1
	4	−28.6010	1	−21.5442	1	−21.0090	1	−109.4157	1	−27.9883	1	−33.2673	1
	5	−5.8490	1	−10.7407	1	−33.9960	1	−110.0810	1	−22.7729	1	−5.4471	1
House	2	−9.5851	1	−11.7822	1	−9.4314	1	−64.9566	1	−11.9011	1	−14.6542	1
	3	−20.6436	1	−15.8776	1	−18.2415	1	−133.0959	1	−22.5299	1	−20.7056	1
	4	−14.1373	1	−19.2192	1	−27.5020	1	−45.3072	1	−28.2547	1	−11.2347	1
	5	−35.3391	1	−22.4042	1	−27.7047	1	−120.2696	1	−38.4260	1	−49.8531	1

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

Table 14
Statistical analysis of 30 runs for the experiments on the between-class variance criterion. *t*-Value: probability value of the *t*-test, $h = 0$ indicates null hypothesis cannot be rejected at the 0.05 significance level, $h = 1$ indicates null hypothesis can be rejected.

Image	*m*	GSA-GA VS GSA		GSA-GA VS PSOGSA		GSA-GA VS GGSA		GSA-GA VS DS		GSA-GA VS BBO-DE		GSA-GA VS GAPSO	
		t-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*	*t*-Value	*h*
Lena	2	−134.9436	1	−59.6239	1	−61.3441	1	−34.6854	1	−75.4565	1	−144.5626	1
	3	−70.4291	1	−49.5220	1	−42.2240	1	−17.4245	1	−229.5995	1	−54.9230	1
	4	−91.4593	1	−40.0704	1	−63.3693	1	−59.3647	1	−166.4417	1	−34.8308	1
	5	−77.5428	1	−36.4134	1	−61.0548	1	−93.6657	1	−245.6113	1	−73.6525	1
Cameraman	2	−34.1211	1	−35.5596	1	−46.0125	1	−53.3167	1	−212.2025	1	−66.4492	1
	3	−39.5100	1	−30.9383	1	−32.1778	1	−55.1051	1	−74.4942	1	−98.8943	1
	4	−4.0221	1	−4.7708	1	−3.8933	1	−3.0423	1	−30.2209	1	−5.4935	1
	5	−23.5911	1	−25.4627	1	−29.6616	1	−35.7725	1	−104.3268	1	−42.5095	1
Butterfly	2	−38.1244	1	−45.8599	1	−26.6132	1	−44.0180	1	−155.6620	1	−47.9347	1
	3	−16.4803	1	−1.5982	0	−1.9646	0	−4.4029	1	−25.2243	1	−2.3782	1
	4	−33.1981	1	−34.6537	1	−33.3674	1	−41.0991	1	−154.7331	1	−42.1022	1
	5	−8.7729	1	−28.5879	1	−59.8262	1	−41.9261	1	−128.5929	1	−42.5990	1
Landscape	2	−101.9428	1	−34.5141	1	−18.9065	1	−57.8202	1	−73.2788	1	−44.6554	1
	3	−40.2312	1	−18.8167	1	−15.2076	1	−20.3754	1	−142.9756	1	−22.2181	1
	4	−48.7220	1	−14.5803	1	−18.6978	1	−26.0543	1	−91.8095	1	−25.2936	1
	5	−57.0470	1	−27.3630	1	−20.8621	1	−25.0061	1	−100.5211	1	−25.3120	1
Starfish	2	−33.8867	1	−18.7511	1	−17.1012	1	−28.7643	1	−98.1611	1	−22.7134	1
	3	−24.9675	1	−14.3605	1	−14.2566	1	−24.9593	1	−84.1085	1	−18.9882	1
	4	−60.5140	1	−33.9114	1	−24.1834	1	−35.4532	1	−166.9768	1	−50.2908	1
	5	−28.6609	1	−33.4624	1	−59.9193	1	−9.6001	1	−119.5735	1	−32.2054	1
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

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054



(a) gray-level histogram of the House image, m=2 (b) Segmented House image, $m = 2$ (c) gray-level histogram of the House image, m=2 (d) Segmented House image, $m = 2$

(e) gray-level histogram of the House image, m=3 (f) Segmented House image, $m = 3$ (g) gray-level histogram of the House image, m=3 (h) Segmented House image, $m = 3$

(i) gray-level histogram of the House image, m=4 (j) Segmented House image, $m = 4$ (k) gray-level histogram of the House image, m=4 (l) Segmented House image, $m = 4$

(m) gray-level histogram of the House image, m=5 (n) Segmented House image, $m = 5$ (o) gray-level histogram of the House image, m=5 (p) Segmented House image, $m = 5$

Fig. 11. The gray-level histogram labeled with the best threshold values and the corresponding segmentation image of the House image by GSA-GA with m varies from 2 to 5. The first and the second columns were based on the entropy criterion. The third and fourth columns were based on the between-class variance criterion.

significantly outperformed DS, BBO-DE, and GAPSO when taking the between-class variance as evaluation criteria as shown in Table 14. Moreover, although the CPU times of GSA-GA were not significantly lower than GGSA and PSOGSA on image Butterfly when $m = 3$, GSA-GA outperformed GGSA and PSOGSA significantly on all the other thresholds.

5. Conclusion

In this paper, we presented a novel multi-level thresholding algorithm for image segmentation by employing an improved GSA variant, called GSA-GA. In GSA, the gravitational force attracts all particles move to those heavier particles. This property causes GSA

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

to trap in local optima and to slow convergence. On the other hand, GSA is time consuming because all the particles are involved when calculating the acceleration of each particle. Therefore, when applying GSA to multi-level image thresholding, it is easy to generate false local optimal thresholds and suffer from high computational complexity. This paper introduced the roulette selection and discrete mutation operators of GA to escape from local optima, and thus improve the search accuracy and speed of GSA. To evaluate the validity of GSA-GA, it was employed on six test images using four different of thresholds with entropy and between-class variance criteria. The graphical and statistical experimental results confirmed the superiority of GSA-GA compared with GSA, PSOGSA, GGSA, DS, BBO-DE, and GAPSO. In addition, the Student's *t*-test showed that GSA-GA significantly improves the computational complexity.

Acknowledgements

The authors are thankful to editors and anonymous reviewers, whose detailed comments and suggestions have notably helped improve the manuscript. This work was supported by Chinese Natural Science Foundation Projects (41471353, 41271349), Fundamental Research Funds for the Central Universities (14CX02039A, 15CX06001A).

Appendix A. Abbreviations used in this paper

Abbreviations	Full name
ABC	Artificial bee colony optimization
ACO	Ant colony optimization
ACO/PSO	Hybrid ant colony optimization with particle swarm optimization
BBO	Biogeography-based optimization
BBO-DE	Hybrid differential evolution and biogeography-based optimization
CFO	Central force optimization
DE	Differential evolution
DS	Differential search algorithm
GA	Genetic algorithm
GAPSO	Hybrid genetic algorithm and particle swarm optimization
GGSA	Adaptive gbest-guided gravitational search algorithm
GSA	Gravitational search algorithm
GSA-GA	Hybrid algorithm of gravitational search algorithm with genetic algorithm
PSO	Particle swarm optimization
PSOGSA	Hybrid particle swarm optimization and gravitational search algorithm
SA	Simulated annealing
SA/PSO	Hybrid simulated annealing with particle swarm optimization
VS	Versus

Appendix B. Detailed definition of symbols

Symbols	Detailed definition	Symbols	Detailed definition
a_{id}^t	Acceleration of the i-th particle in the d-th dimension at time t	p_1 & p_2	Two control parameters in DS
BaseM	Mutation mask matrix	p_c	Crossover probability
$best^t$	The best fitness value of the population at time t	$p_{cs}^t(i)$	Cumulative probability of the particle i at time t
c_1& c_2	Two acceleration coefficients in GGSA	p_m	Mutation probability
ceil	A function for computing round toward positive infinity	$p_s^t(i)$	Selection probability of the particle i
CR	Crossover probability in BBO-DE	$\boldsymbol{pbest}_i$	The best experience of the i-th particle itself
d	Sequence number of the d-th component in an array	$prob_i$	Probability of the gray-level i
F	Scaling factor in BBO-DE	R_{ij}^t	Euclidian distance between particles i and j at time t
f_i	Gray-level of pixel i	*rand*	Uniform random variable in the interval [0, 1]
F_{index}	Index of thresholding criterion	Re_j	The j-th segmented region
$F_i^d(t)$	Total force acts on the particle i in the d-th dimension at time t	*rem*	A function for computing the remainder after division
$F_{ij}^d(t)$	Force acts on the particle i from the particle j at time t	S	Total dimension of the search space
f_{max}	Maximum gray-level of pixels in the given image	*std*(***fit***)	Standard deviation of the population's fitness
f_{min}	Minimum gray-level of pixels in the given image	*Std_t*	Stand deviation of CPU times
FEs	Maximum number of fitness evaluations	*Std_u*	Stand deviation of uniformity
fit	Fitness matrix of all particles	t	The current iteration/time
fit_i^t	Fitness value of the i-th particle at time t	td_i	The value of the i-th threshold
fit_j^t	Fitness value of the j-th particle at time t	*Th*	Obtained threshold values
G	Gravitational constant	t-Value	Probability value of the t-test
G_0	Initial gravitational constant	u	Uniformity measure
$G(t)$	Gravitational constant at time t	$\boldsymbol{v}$	Velocity matrix of the population
g_j	Mean gray-level of pixels in j-th segmented region	$\boldsymbol{v}_i$	Velocity of the i-th particle
gbest	The best experience of the population	v_{id}	Velocity of the i-th particle in the d-th dimension
h	Indicates whether the null hypothesis is rejected	v_{id}^t	Velocity of the i-th particle in the d-th dimension at time t
HE_i	Entropy of class i	v_{id}^{t+1}	Velocity of the i-th particle in the d-th dimension at time $t+1$
$hist_i$	Number of pixels with gray-level i	$worst^t$	The worst fitness value of the population at time t
i	Sequence number of the i-th component in an array	$\boldsymbol{x}$	Position matrix of the population
IMG	The given image	$\boldsymbol{x}_i$	Position of the i-th particle
$Iter_{\max}$	Maximum number of iterations	$\boldsymbol{x}_i^{t+1}$	Position matrix of the population at time t+1
j	Sequence number of the j-th component in an array	x_{id}	Position of the i-th particle in the d-th dimension

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

L	Maximum gray-levels of the given image	x_{id}^t	Position of the i-th particle in the d-th dimension at time t
m	The required number of thresholds	x_{id}^{t+1}	Position of the i-th particle in the d-th dimension at time $t+1$
$Mass_i^t$	Mass of the particle i at time t	$\boldsymbol{x}_j^t$	Position of the i-th particle at time t
$Mass_j^t$	Mass of the particle j at time t	x_{jd}^t	Position of the j-th particle in the d-th dimension at time t
M_t	Mean of CPU times	α	Significance level
M_u	Mean of uniformity	β	Decrease coefficient
N	The size of population $\boldsymbol{x}$	ε	Small constant which is bigger than 0
n_{elit}	Nelit elitism parameter in BBO-DE	σ_i	Sigma functions of class i
$\boldsymbol{Newx}_i^t$	Position of the i-th particle selected by roulette selection	μ_i	Mean gray-level of class i
$nmfit_i^t$	Normalized fitness value	μ_T	Mean intensity of the given image
Num	Total number of pixels in the given image	ω_i	Probability of class i

References

[1] Y. Zhang, D. Huang, M. Ji, F. Xie, Image segmentation using PSO and PCM with Mahalanobis distance, Expert Syst. Appl. 38 (7) (2011) 9036–9040, http://dx.doi.org/10.1016/j.eswa.2011.01.041.
[2] M.-H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Syst. Appl. 38 (11) (2011) 13785–13791, http://dx.doi.org/10.1016/j.eswa.2011.04.180.
[3] P. Ghamisi, M.S. Couceiro, F.M. Martins, J. Atli Benediktsson, Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization, IEEE Trans. Geosci. Remote Sens. 52 (5) (2014) 2382–2394, http://dx.doi.org/10.1109/TGRS.2013.2260552.
[4] J.N. Kapur, P.K. Sahoo, A.K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Gr. Image Process. 29 (3) (1985) 273–285, http://dx.doi.org/10.1016/0734-189X(85)90125-2.
[5] S. Sarkar, S. Das, S.S. Chaudhuri, A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution, Pattern Recognit. Lett. 54 (2015) 27–35, http://dx.doi.org/10.1016/j.patrec.2014.11.009.
[6] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66, http://dx.doi.org/10.1109/TSMC.1979.4310076.
[7] B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. 13 (6) (2013) 3066–3091, http://dx.doi.org/10.1016/j.asoc.2012.03.072.
[8] X. Li, Z. Zhao, H. Cheng, Fuzzy entropy threshold approach to breast cancer detection, Inf. Sci. Appl. 4 (1) (1995) 49–56, http://dx.doi.org/10.1016/1069-0115(94)00019-X.
[9] J. Kittler, J. Illingworth, Minimum error thresholding, Pattern Recognit. 19 (1) (1986) 41–47, http://dx.doi.org/10.1016/0031-3203(86)90030-0.
[10] T. Kurban, P. Civicioglu, R. Kurban, E. Besdok, Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding, Appl. Soft Comput. 23 (2014) 128–143, http://dx.doi.org/10.1016/j.asoc.2014.05.037.
[11] A. Chander, A. Chatterjee, P. Siarry, A new social and momentum component adaptive PSO algorithm for image segmentation, Expert Syst. Appl. 38 (5) (2011) 4998–5500, http://dx.doi.org/10.1016/j.eswa.2010.09.151.
[12] M. Ali, C.W. Ahn, M. Pant, Multi-level image thresholding by synergetic differential evolution, Appl. Soft Comput. 17 (2014) 1–11, http://dx.doi.org/10.1016/j.asoc.2013.11.018.
[13] E.L. Lawler, D.E. Wood, Branch-and-bound methods: a survey, Oper. Res. 14 (4) (1966) 699–719, http://dx.doi.org/10.1287/opre.14.4.699.
[14] F. Glover, G.A. Kochenberger, Handbook of Metaheuristics, Springer Science & Business Media, 2003.
[15] J. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-based Algorithms, vol. 97, Springer Science & Business Media, 2005.
[16] S. Kirkpatrick, Optimization by simulated annealing: quantitative studies, J. Stat. Phys. 34 (5–6) (1984) 975–986, http://dx.doi.org/10.1007/BF01009452.
[17] M. origo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperation agents, IEEE Trans. Syst. Man Cybern. B: Cybern. 26 (1) (1996) 29–41, http://dx.doi.org/10.1109/3477.484436.
[18] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. rep., Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[19] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.
[20] I. Boussaïd, A. Chatterjee, P. Siarry, M. Ahmed-Nacer, Hybrid BBO-DE algorithms for fuzzy entropy-based thresholding, in: Computational Intelligence in Image Processing, Springer, 2013, pp. 37–69, http://dx.doi.org/10.1007/978-3-642-30621-1_3.
[21] P. Civicioglu, Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm, Comput. Geosci. 46 (2012) 229–247, http://dx.doi.org/10.1016/j.cageo.2011.12.011.
[22] J. Kenndy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[23] E.A. Baniani, A. Chalechale, Hybrid PSO and genetic algorithm for multilevel maximum entropy criterion threshold selection, Int. J. Hybrid Inf. Technol. 6 (5) (2013) 131–140, http://dx.doi.org/10.14257/ijhit.2013.6.5.12.
[24] C.-F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Trans. Syst. Man Cybern. B: Cybern. 34 (2) (2004) 997–1006, http://dx.doi.org/10.1109/TSMCB.2003.818557.
[25] M.K. Patel, M.R. Kabat, C.R. Tripathy, A hybrid ACO/PSO based algorithm for QOS multicast routing problem, Ain Shams Eng. J. 5 (1) (2014) 113–120, http://dx.doi.org/10.1016/j.asej.2013.07.005.
[26] Y. Zhang, L. Wu, A robust hybrid restarted simulated annealing particle swarm optimization technique, Adv. Comput. Sci. Appl. 1 (1) (2012) 5–8.
[27] P.-Y. Yin, A fast scheme for optimal thresholding using genetic algorithms, Signal Process. 72 (2) (1999) 85–95, http://dx.doi.org/10.1016/S0165-1684(98)00167-4.
[28] W.-B. Tao, J.-W. Tian, J. Liu, Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm, Pattern Recognit. Lett. 24 (16) (2003) 3069–3078, http://dx.doi.org/10.1016/S0167-8655(03)00166-1.
[29] K. Hammouche, M. Diaf, P. Siarry, A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation, Comput. Vis. Image Underst. 109 (2) (2008) 163–175, http://dx.doi.org/10.1016/j.cviu.2007.09.001.
[30] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73, http://dx.doi.org/10.1109/4235.985692.
[31] P.-Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, Appl. Math. Comput. 184 (2) (2007) 503–513, http://dx.doi.org/10.1016/j.amc.2006.06.057.
[32] S. Nabizadeh, K. Faez, S. Tavassoli, A. Rezvanian, A novel method for multilevel image thresholding using particle swarm optimization algorithms, in: Proceedings of 2010 2nd International Conference on Computer Engineering and Technology, vol. 4, IEEE, 2010, p. V4-271, http://dx.doi.org/10.1109/ICCET.2010.5485600.
[33] H.V.H. Ayala, F.M. dos Santos, V.C. Mariani, L. dos Santos Coelho, Image thresholding segmentation based on a novel beta differential evolution approach, Expert Syst. Appl. 42 (4) (2015) 2136–2142, http://dx.doi.org/10.1016/j.eswa.2014.09.043.
[34] Z. Beheshti, S.M.H. Shamsuddin, S. Hasan, MPSO: median-oriented particle swarm optimization, Appl. Math. Comput. 219 (11) (2013) 5817–5836, http://dx.doi.org/10.1016/j.ins.2013.08.015.
[35] Z. Beheshti, S.M.H. Shamsuddin, CAPSO: centripetal accelerated particle swarm optimization, Inf. Sci. 258 (2014) 54–79, http://dx.doi.org/10.1016/j.ins.2013.08.015.
[36] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417, http://dx.doi.org/10.1109/TEVC.2008.927706.
[37] M. Hu, T.-F. Wu, J.D. Weir, An adaptive particle swarm optimization with multiple adaptive methods, IEEE Trans. Evol. Comput. 17 (5) (2013) 705–720, http://dx.doi.org/10.1109/TEVC.2012.2232931.
[38] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248, http://dx.doi.org/10.1016/j.ins.2009.03.004.
[39] S. Jiang, Y. Wang, Z. Ji, Convergence analysis and performance of an improved gravitational search algorithm, Appl. Soft Comput. 24 (2014) 363–384, http://dx.doi.org/10.1016/j.asoc.2014.07.016.
[40] A. Zhang, G. Sun, Z. Wang, Y. Yao, A hybrid genetic algorithm and gravitational search algorithm for global optimization, Neural Netw. World 25 (1) (2015) 53, http://dx.doi.org/10.14311/NNW.2015.25.003.
[41] S. Mirjalili, S.Z.M. Hashim, H.M. Sardroudi, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, Appl. Math. Comput. 218 (22) (2012) 11125–11137, http://dx.doi.org/10.1016/j.amc.2012.04.069.
[42] J.V. Kumar, D.V. Kumar, K. Edukondalu, Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market, Appl. Soft Comput. 13 (5) (2013) 2445–2455, http://dx.doi.org/10.1016/j.asoc.2012.12.003.
[43] N.M. Sabri, M. Puteh, M.R. Mahmood, A review of gravitational search algorithm, Int. J. Adv. Soft Comput. 5 (3) (2013) 1–39.
[44] S. Sarafrazi, H. Nezamabadi-Pour, S. Saryazdi, Disruption: a new operator in gravitational search algorithm, Scientia Iranica 18 (3) (2011) 539–548, http://dx.doi.org/10.1016/j.scient.2011.04.003.

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054

[45] S. Mirjalili, S.Z.M. Hashim, A new hybrid PSOGSA algorithm for function optimization, in: Proceedings of 2010 International Conference on Computer and Information Application, IEEE, 2010, pp. 374–377, http://dx.doi.org/10.1109/ICCIA.2010.6141614.
[46] S. Mirjalili, A. Lewis, Adaptive gbest-guided gravitational search algorithm, Neural Comput. Appl. 25 (7–8) (2014) 1569–1584, http://dx.doi.org/10.1007/s00521-014-1640-y.
[47] X. Han, X. Chang, A chaotic digital secure communication based on a modified gravitational search algorithm filter, Inf. Sci. Int. J. 208 (21) (2012) 14–27, http://dx.doi.org/10.1016/j.ins.2012.04.039.
[48] F. Herrera, M. Lozano, J.L. Verdegay, Fuzzy connectives based crossover operators to model genetic algorithms population diversity, Fuzzy Sets Syst. 92 (1) (1997) 21–30, http://dx.doi.org/10.1016/S0165-0114(96)00179-0.
[49] Y.V. Pehlivanoglu, A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks, IEEE Trans. Evol. Comput. 17 (3) (2013) 436–452, http://dx.doi.org/10.1109/TEVC.2012.2196047.
[50] W. Gong, Z. Cai, C.X. Ling, H. Li, A real-coded biogeography-based optimization with mutation, Appl. Math. Comput. 216 (9) (2010) 2749–2758, http://dx.doi.org/10.1016/j.amc.2010.03.123.
[51] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, U Michigan Press, 1975.
[52] H.M. Azamathulla, F.-C. Wu, A. Ab Ghani, S.M. Narulkar, N.A. Zakaria, C.K. Chang, Comparison between genetic algorithm and linear programming approach for real time operation, J. Hydro-environ. Res. 2 (3) (2008) 172–181, http://dx.doi.org/10.1016/j.jher.2008.10.001.
[53] F. Wang, Y. Qiu, A modified particle swarm optimizer with roulette selection operator, in: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, IEEE, 2005, pp. 765–768, http://dx.doi.org/10.1109/NLPKE.2005.1598839.
[54] P. Narain, Genetic variability under step-wise discrete mutation and stabilizing selection, J. Indian Soc. Agric. Stat. 44 (2) (1992), 171–171.
[55] G. Sun, A. Zhang, A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding, in: Pattern Recognition and Image Analysis, Springer, 2013, pp. 707–714, http://dx.doi.org/10.1007/978-3-642-38628-2_84.
[56] A. Pizurica, W. Philips, I. Lemahieu, M. Acheroy, A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising, IEEE Trans. Image Process. 11 (5) (2002) 545–557, http://dx.doi.org/10.1109/TIP.2002.1006401.
[57] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of 2001 Eighth IEEE International Conference on Computer Vision, vol. 2, IEEE, 2001, pp. 416–423, http://dx.doi.org/10.1109/ICCV.2001.937655.
[58] P.K. Sahoo, S. Soltani, A.K. Wong, A survey of thresholding techniques, Comput. Vis. Gr. Image Process. 41 (2) (1988) 233–260, http://dx.doi.org/10.1016/0734-189X(88)90022-9.
[59] Y. Li, L. Jiao, R. Shang, R. Stolkin, Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation, Inf. Sci. 294 (2015) 408–422, http://dx.doi.org/10.1016/j.ins.2014.10.005.
[60] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295, http://dx.doi.org/10.1109/TEVC.2005.857610.
[61] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.

Please cite this article in press as: G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.054